Getting Started

About NeuralForecast

NeuralForecast offers a large collection of neural forecasting models focused on their usability, and robustness. The models range from classic networks like `MLP`, `RNN`s to novel proven contributions like `NBEATS`, `NHITS`, `TFT` and other architectures.

🎊 Features

- **Exogenous Variables**: Static, historic and future exogenous support.
- **Forecast Interpretability**: Plot trend, seasonality and exogenous `_NBEATS_`, `_NHITS_`, `_TFT_`, `ESRNN` prediction components.
- **Probabilistic Forecasting**: Simple model adapters for quantile losses and parametric distributions.
- **Train and Evaluation Losses** Scale-dependent, percentage and

scale independent errors, and parametric likelihoods.

- **Automatic Model Selection** Parallelized automatic hyperparameter tuning, that efficiently searches best validation configuration.
- **Simple Interface** Unified SKLearn Interface for `StatsForecast` and `MLForecast` compatibility.
- **Model Collection**: Out of the box implementation of `_MLP_`, `_LSTM_`, `_RNN_`, `_TCN_`, `_DilatedRNN_`, `_NBEATS_`, `_NHITS_`, `ESRNN`, `_Informer_`, `_TFT_`, `_PatchTST_`, `_VanillaTransformer_`, `_StemGNN_` and `_HINT_`. See the entire **collection here**.

Why?

There is a shared belief in Neural forecasting methods' capacity to improve our pipeline's accuracy and efficiency.

Unfortunately, available implementations and published research are yet to realize neural networks' potential. They are hard to use and continuously fail to improve

over statistical methods while being computationally prohibitive. For this reason, we created `NeuralForecast`, a library favoring proven accurate and efficient models focusing on their usability.

💻 Installation

PyPI

You can install `NeuralForecast`'s *released version* from the Python package index **pip** with:

```
pip install neuralforecast
```

(Installing inside a python virtualenvironment or a conda environment is recommended.)

Conda

Also you can install `NeuralForecast`'s *released version* from **conda** with:

```
conda install -c conda-forge neura
```

(Installing inside a python virtualenvironment or a conda

environment is recommended.)

Dev Mode

If you want to make some modifications to the code and see the effects in real time (without reinstalling), follow the steps below:

```
git clone https://github.com/Nixtl
cd neuralforecast
pip install -e .
```

How to Use

```
import logging

import pandas as pd
from utilsforecast.plotting import
from neuralforecast import NeuralF
from neuralforecast.models import
from neuralforecast.utils import A
```

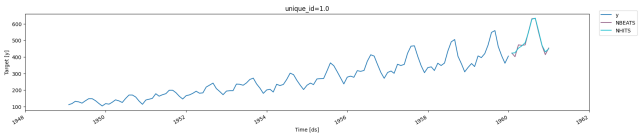
```
logging.getLogger('pytorch_lightni
```

```
# Split data and declare panel dat
Y_df = AirPassengersDF
Y_train_df = Y_df[Y_df.ds<='1959-1
Y_test_df = Y_df[Y_df.ds>'1959-12-

# Fit and predict with NBEATS and
horizon = len(Y_test_df)
models = [NBEATS(input_size=2 * ho
          NHITS(input_size=2 * hor
nf = NeuralForecast(models=models,
nf.fit(df=Y_train_df)
Y_hat_df = nf.predict()

# Plot predictions
plot_series(Y_train_df, Y_hat_df)
```

```
Seed set to 1
Seed set to 1
```



🙏 How to Cite

If you enjoy or benefit from using these Python implementations, a citation to the repository will be greatly appreciated.

```
@misc{olivares2022library_neuralfo
    author={Kin G. Olivares and
            Cristian Challú and
            Federico Garza and
            Max Mergenthaler Canse
            Artur Dubrawski},
    title = {{NeuralForecast}: Use
    year={2022},
    howpublished={{PyCon} Salt Lak
    url={https://github.com/Nixtla
}
```

Was this page helpful? Yes No Suggest edits Raise issue

Quickstart ›