**Open Source Libraries for Time Series Forecasting**

# Overview

Time series forecasting is a method in the statistics field to analyse historical data with a time component and create a prediction based on it.

Some classic examples of time series forecasting methods are Moving Average, ARIMA, and Exponential Smoothing. These methods have been used for a long time and are still useful now because of how easy it is for users to explain the result — although with less accurate prediction.

In turn, many machine learning-driven forecasts are developed by sacrificing some of the explainability and improving accuracies. Whether you need a classical approach or a machine learning-driven model, there are several Python packages to access all these methods in an automated fashion.

Following are a few of the useful packages to accelerate the development time:

# Auto\_TS

* Automatically build multiple Time Series models using a Single Line of Code. Now updated with Dask.
* Auto\_timeseries is a complex model building utility for time series data. Since it automates many Tasks involved in a complex endeavor, it assumes many intelligent defaults. But you can change them. Auto\_Timeseries will rapidly build predictive models based on Statsmodels ARIMA, Seasonal ARIMA, Prophet and Scikit-Learn ML. It will automatically select the best model which gives best score specified.
* New version 0.0.35 onwards has major updates: You can now load your file into Dask dataframes. Just provide the name of your file and if it is too large to fit into a pandas dataframe, Auto\_TS will automatically detect and load it into a Dask dataframe.
* Also, new since version 0.0.25 is the syntax of Auto\_TimeSerie: It is now more like scikit-learn (with fit and predict). You will have to initialize an object and then call fit with your data and then predict again with data. Hope this makes it easier to remember and use
* Github link: <https://github.com/AutoViML/Auto_TS>

# Darts

* Darts is a Python library for user-friendly forecasting and anomaly detection on time series. It contains a variety of models, from classics such as ARIMA to deep neural networks. The forecasting models can all be used in the same way, using fit() and predict() functions, similar to scikit-learn.
* The library also makes it easy to backtest models, combine the predictions of several models, and take external data into account. Darts supports both univariate and multivariate time series and models.
* The ML-based models can be trained on potentially large datasets containing multiple time series, and some of the models offer a rich support for probabilistic forecasting.
* Darts also offers extensive anomaly detection capabilities. For instance, it is trivial to apply PyOD models on time series to obtain anomaly scores, or to wrap any of Darts forecasting or filtering models to obtain fully fledged anomaly detection models.
* Github link: <https://github.com/unit8co/darts>

# Kats

* Kats is a toolkit to analyze time series data, a lightweight, easy-to-use, and generalizable framework to perform time series analysis.
* Time series analysis is an essential component of Data Science and Engineering work at industry, from understanding the key statistics and characteristics, detecting regressions and anomalies, to forecasting future trends.
* Kats aims to provide the one-stop shop for time series analysis, including detection, forecasting, feature extraction/embedding, multivariate analysis, etc.
* Kats is released by Facebook's *Infrastructure Data Science* team. It is available for download on [PyPI](https://pypi.python.org/pypi/kats/)
* Github link: <https://github.com/facebookresearch/kats>

# Nixtla

* Open-source time-series pipeline capable of achieving 1% of the performance in the M5 competition.
* Our open-source solution has a 25% better accuracy than Amazon Forecast and is 20% more accurate than fbprophet.
* It also performs 4x faster than Amazon Forecast and is less expensive
* Features include: tspreprocess to preprocess time-series data such as missing values imputation; tsfeatures to generate features to include in the models; tsforecast to perform forecast at scale; tsbenchmarks to easily calculate accuracy baselines.

NeurosalForecast for state of the art deep learning models

* Github link: <https://github.com/Nixtla/nixtla>

# Pmdarima

* Pmdarima (originally pyramid-arima, for the anagram of 'py' + 'arima') is a statistical library designed to fill the void in Python's time series analysis capabilities. This includes:
  + The equivalent of R's auto.arima functionality
  + A collection of statistical tests of stationarity and seasonality
  + Time series utilities, such as differencing and inverse differencing
  + Numerous endogenous and exogenous transformers and featurizers, including Box-Cox and Fourier transformations
  + Seasonal time series decompositions
  + Cross-validation utilities
  + A rich collection of built-in time series datasets for prototyping and examples
  + Scikit-learn-esque pipelines to consolidate your estimators and promote productionization
* Pmdarima wraps statsmodels under the hood, but is designed with an interface that's familiar to users coming from a scikit-learn background.
* Github link: <https://github.com/alkaline-ml/pmdarima>

# Scalecast

* Scalecast helps you forecast time series. What sets it apart from other libraries is its pipelining functionality.
* The unique approach not only allows a series to be transformed to account for stationarity and other concerns, but also fully reverted when results are ready to be reported. Point forecasts, test-set metrics, and conformal confidence intervals are easily obtained at the original series level through this process.
* Uniform ML modeling (with models from a diverse set of libraries, including scikit-learn, statsmodels, and tensorflow), reporting, and data visualizations are offered through the Forecaster and MVForecaster interfaces.
* Data storage and processing then becomes easy as all applicable data, predictions, and many derived metrics are contained in a few objects with much customization available through different modules.
* Github link: <https://github.com/mikekeith52/scalecast>

# Skforecast

* Time series forecasting with scikit-learn regressors.
* Skforecast is a python library that eases using scikit-learn regressors as multi-step forecasters. It also works with any regressor compatible with the scikit-learn API (pipelines, CatBoost, LightGBM, XGBoost, Ranger...).
* Why use skforecast?
  + Skforecast is developed according to the following priorities:
  + Fast and robust prototyping.
  + Validation and backtesting methods to have a realistic assessment of model performance.
  + Models must be deployed in production.
  + Models must be interpretable.
* Github link: <https://github.com/JoaquinAmatRodrigo/skforecast>

# Sktime

* sktime is a library for time series analysis in Python. It provides a unified interface for multiple time series learning tasks.
* Currently, this includes time series classification, regression, clustering, annotation and forecasting. It comes with time series algorithms and scikit-learn compatible tools to build, tune and validate time series models.
* Github link: <https://github.com/sktime/sktime>

# Statsforecast

* Lightning fast forecasting with statistical and econometric models
* StatsForecast offers a collection of widely used univariate time series forecasting models, including automatic ARIMA, ETS, CES, and Theta modeling optimized for high performance using numba. It also includes a large battery of benchmarking models
* Github link: <https://github.com/Nixtla/statsforecast>

# GluonTS

* GluonTS is a Python package for probabilistic time series modeling, focusing on deep learning based models, based on PyTorch and MXNet.
* Github link: <https://github.com/awslabs/gluonts>