A picture containing logo

Description automatically generated

Amazon Forecast

comparison

[rks](https://www.tangent.works)

Amazon Forecast is a time-series forecasting service based on machine learning (ML). It provides six algorithms to predict future time-series data based on historical data and requires no machine learning experience.

There is a possibility (recommended) to set the **AutoML** (AutoPredictor) approach, which will find the best algorithm/model (Predictor) automatically.

# Using Amazon Forecast

To import time-series datasets, train predictors, and generate forecasts, you can use the following:

Canvas (visual interface), Amazon Forecast console, [Python Software Development Kit](https://docs.aws.amazon.com/forecast/latest/dg/getting-started-python.html) (SDK), AWS Command Line Interface(AWS CLI) and APIs.

# How it works

A picture containing chart

Description automatically generated

When creating forecasting projects in Amazon Forecast, you work with the following resources:

* [**Importing Datasets**](https://docs.aws.amazon.com/forecast/latest/dg/howitworks-datasets-groups.html)– Datasets are collections of your input data. Forecast algorithms use your data to train custom forecasting models, called predictors.
* [**Training Predictors**](https://docs.aws.amazon.com/forecast/latest/dg/howitworks-predictor.html)– Predictors are custom models trained on your data. You can train a predictor by choosing a prebuilt algorithm or choosing the AutoML option to have Amazon Forecast pick the best algorithm for you.
* [**Generating Forecasts**](https://docs.aws.amazon.com/forecast/latest/dg/howitworks-forecast.html)– You can generate forecasts for your time-series data and query them.
* **Explore Insights-** Attributes impact/ What-if analysis**.**

# Important observations

To understand the possibilities and limits of Amazon Forecast.

### Import data

It is possible to include the predictors/features (related time-series), but it is constrained as they can come only in two forms:

* **Historical time series**: historical related time series must contain data points up to the forecast horizon
* **Forward-looking time series:** forward-looking related time series must contain data points up to and within the whole forecast horizon.

Chart

Description automatically generated with medium confidence

In addition :

* related time series must start on or before the beginning timestamp of the target time series
* related time series cannot have missing values
* related time series dataset can contain only up to 13 related time-series features

Conclusion: Because of these limitations, Amazon Forecast can not be used for many data problems.

NOTE: Even though there are only these related time-series in your data, not all Amazon Forecast algorithms can include them. (more in next section)

## Training predictors

AutoPredictor is the default and preferred method to create a predictor with Amazon Forecast. AutoPredictor creates a model (predictor) by applying the optimal algorithm for your problem.

Building of AutoPredictor **takes from 2 to 4 hours**.

The following table shows the types of related time series each Amazon Forecast algorithm accepts.

Calendar

Description automatically generated

When using AutoML, you can provide both historical and forward-looking related time-series data, but the usage of algorithms is limited :

* Amazon Forecast will only use those time series where applicable
* If you provide forward-looking related time series data, Amazon Forecast will use the related time series data with CNN-QR, DeepAR+, and Prophet and will not use them with NPTS, ARIMA and ETS
* If provided historical related time series data, Amazon Forecast will use the related data with CNN-QR and not use the related data with DeepAR+, Prophet, NPTS, ARIMA, and ETS.

## Generating forecasts

As soon as your model(predictor) is active, you can create a forecast. But the process to get your predictions is very time consuming as you have to :

1. Create forecast ( 15-30 min)
2. Create forecast export – allows you to export forecasts generated by create forecast (1-2 min)

The process is even more tedious if you want to use an existing predictor to generate forecasts with the updated data ( that is often the case as you do not want to retrain your predictor each time the updated dataset is imported) :

1. Upload the updated CSV file to an Amazon S3 bucket. The updated CSV **should still contain all of your existing data** (no time/seconds)
2. Create a dataset import job with the new data. The **most recent import job is the one that forecasts are generated off of** (several minutes or longer – based on the dataset size, it can be 30 minutes)
3. Create a new forecast using the existing predictor. (15-30 min)
4. Retrieve the forecast as usual. (1-2 min)

**NOTE:** Forecsts are generated at the end of the actual data (most recent import job). How many samples ahead are forecast was defined in predictor.

Conclusion:

* As to make a forecast is so time-consuming it is not appropriate for problems when you have to forecast often ( every minute,5min, 10min, 15min up to 1hour)
* Experiments/benchmarks with many rolling windows would be extremely time-consuming as you would have to repeat the above process many times

# Benchmark

## Inventory management

DATA: *Inventory\_Management*

One Target : *Sales*

Related Time Series : *IsOpen,Promo,StateHoliday,SchoolHoliday,TheDayOfWeek*

NOTE: all related time series are available throughout the whole for.horizon ( always available 14 days ahead). It means they are all Forward-looking related time series

Granularity: *Day*

Forecasting horizon: 14 days

Training Period : 2013-01-01 00:00:00 - 2015-01-02 00:00:00

Testing Period : 2015-01-03 00:00:00 - 2015-07-31 00:00:00

Rolling window: 14 days ( every two weeks, we forecast the next two weeks)

## Errors

|  |  |  |
| --- | --- | --- |
|  | MAE | WMAPE |
| TIM | 934.4908551 | 8.435044411 % |
| Amazon Forecast | 1744.280113 | 15.74448818 % |

## Timings

Building a predictor

|  |  |
| --- | --- |
| TIM | 13 s |
| Amazon Forecast | 1h 39 min |

Getting one forecast ( 14 samples)

|  |  |
| --- | --- |
| TIM | 3s |
| Amazon Forecast | 30 min |

To fill the whole out-of-sample period, there was a need to repeat the 14 days ahead forecast 15 times, meaning to repeat the process of updating data, creating a dataset import job, creating a new forecast and creating the forecast export job. By using TIM, you can do it at once ( by setting the out-of-sample period and a corresponding rolling window-in this case, 14)

|  |  |
| --- | --- |
| Amazon Forecast | 7,5 h |

## Electricity price

DATA: *Electricity price*

One Target : *Price*

Related Time Series : *Total load, Zonal load*

NOTE: both related time series are available throughout the whole for.horizon ( always available 24 hours ahead). It means they are all Forward-looking related time series

Granularity: *Hour*

Forecasting horizon: 24 hours

Training Period : 2011-01-01 00:00:00 - 2012-12-31 23:00:00

Testing Period : 2013-01-01 00:00:00 - 2013-01-10 23:00:00

Rolling window: 24hours ( every day at 23:00, we forecast the next day)

## Errors

|  |  |  |
| --- | --- | --- |
|  | MAE | WMAPE |
| TIM | 10.48643376 | 17.09471722 % |
| Amazon Forecast | 16.41343687 | 26.96968009% |

## Timings

Building a predictor

|  |  |
| --- | --- |
| TIM | 50 s |
| Amazon Forecast | 1h 49 min |

Getting one forecast ( 24 samples)

|  |  |
| --- | --- |
| TIM | 4s |
| Amazon Forecast | 30 min |

To fill the whole out-of-sample period, there was a need to repeat the day ahead forecast(24 samples) 10 times, meaning to repeat the process of updating data, creating a dataset import job, creating a new forecast and creating the forecast export job. By using TIM, you can do it at once ( by setting the out-of-sample period and a corresponding rolling window- in this case, 24)

|  |  |
| --- | --- |
| Amazon Forecast | 5 h |