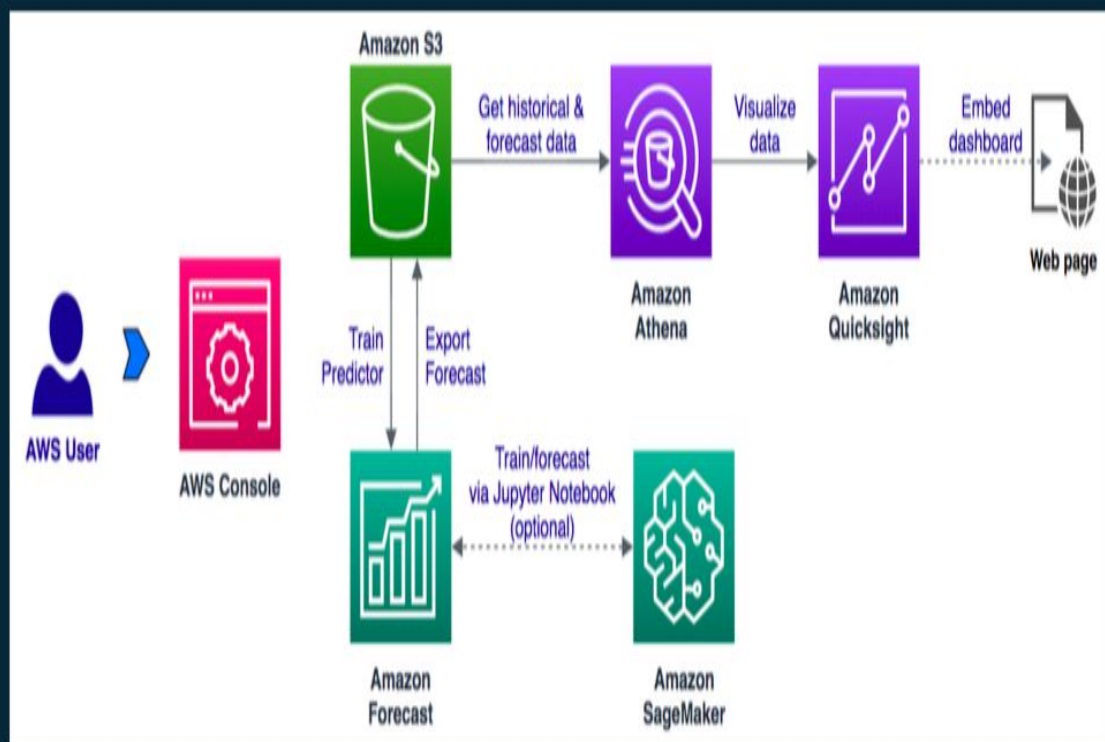


Console Demo Architecture



Amazon Forecast: How it works

Historical data

Sales, inventory, pricing, etc.

Related data

Weather, competitive promotions, etc.

1. Load data
2. Inspect data
3. Identify features
4. Select algorithms
5. Select hyperparameters
6. Train models
7. Optimize models
8. Deploy and host models

Amazon Forecast



Private
Customized
Forecasting
API

Traditional time-series models

- Independent forecasts
- Strong structural assumptions
- De facto industry standard
- Well-understood, >50 yrs. Research
- High data efficiency
- Data must match the structural assumptions
- Cannot identify patterns across time series

Algorithms

- Nonparametric Time Series Model
- Exponential Smoothing (ETS)
- (Auto-) ARIMA
- Prophet



Deep learning time-series models

Global models: identify patterns using all available time series

- Group-dependent seasonality and lifecycle
- Behavior in response to covariate inputs
- Weak structural assumptions
- Can be significantly more accurate than traditional methods
- Can easily incorporate and learn from rich metadata
- Support cold-start forecasts for new items

Algorithms

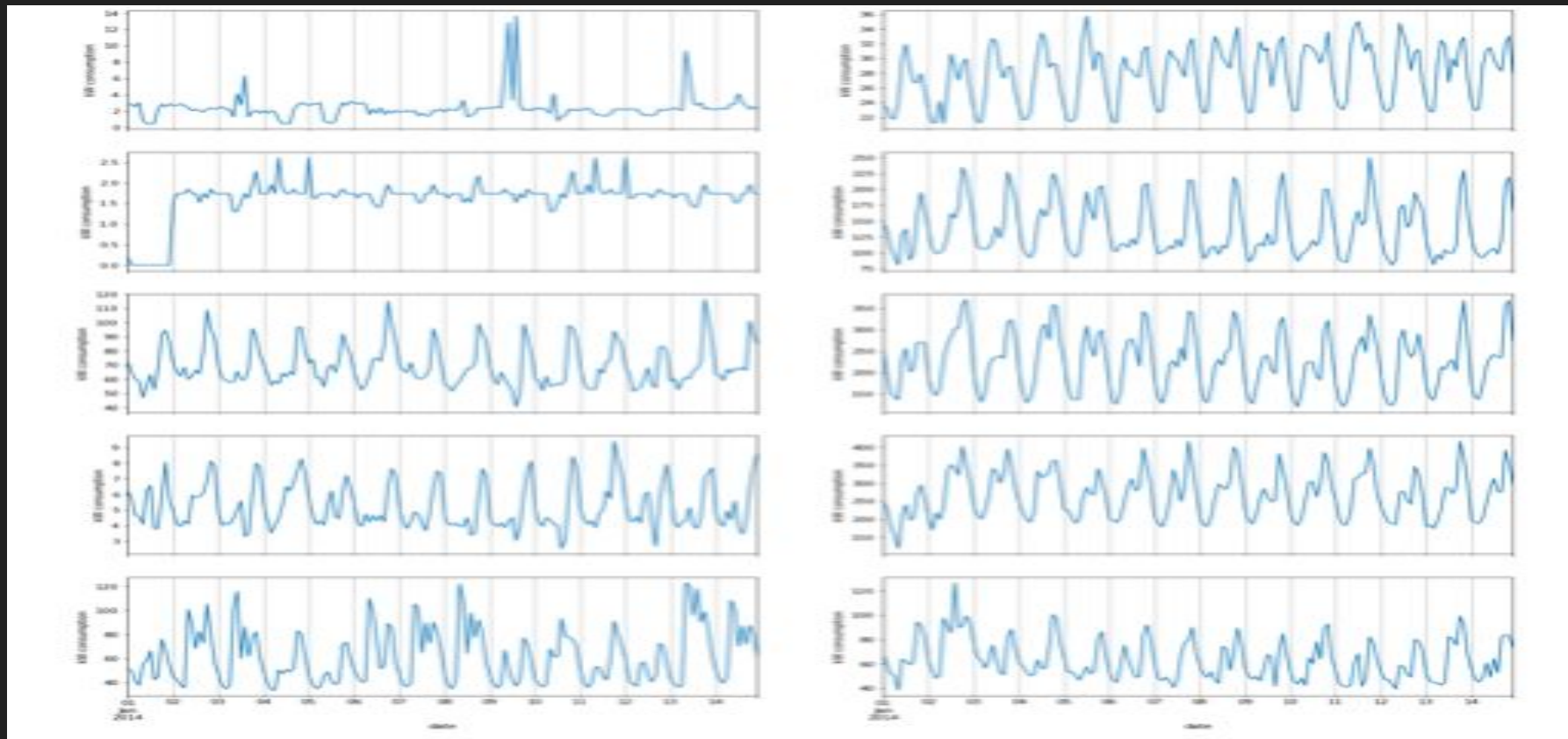


- Autoregressive LSTM (DeepAR)
- Spline Quantile Forecaster (SQF)
- Multi-Horizon Quantile Recurrent Forecaster (MQ-RNN)
- Mixture Density Network RNN (MDN)

Some Amazon Forecast Algorithms

- Auto-Regressive Integrated Moving Average (ARIMA)
- Exponential Smoothing (ETS)
- Non-Parametric Time Series (NPTS)
- Prophet
- Deep Auto-Regressive Plus (DeepAR+) – *Supports HPO*

Plot the resulting time series for the first ten customers for the time period spanning the first two weeks of 2014.



Perform predictions by deploying it to an endpoint

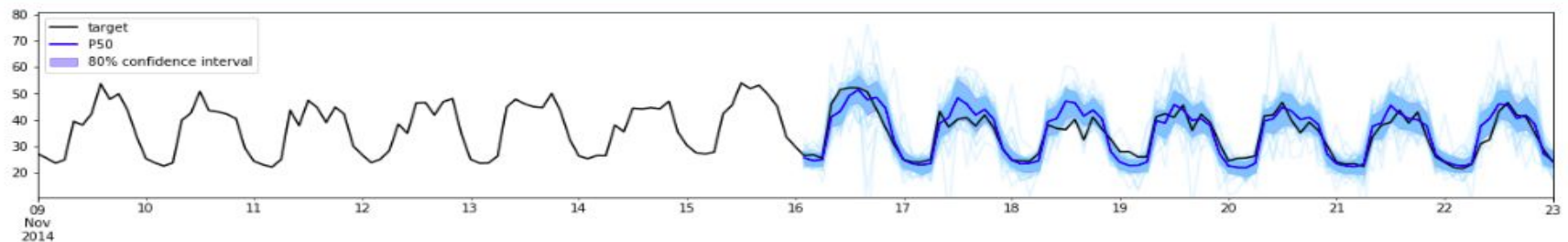
customer_id 119
forecast_day 76
confidence 80
history_weeks_plot 1

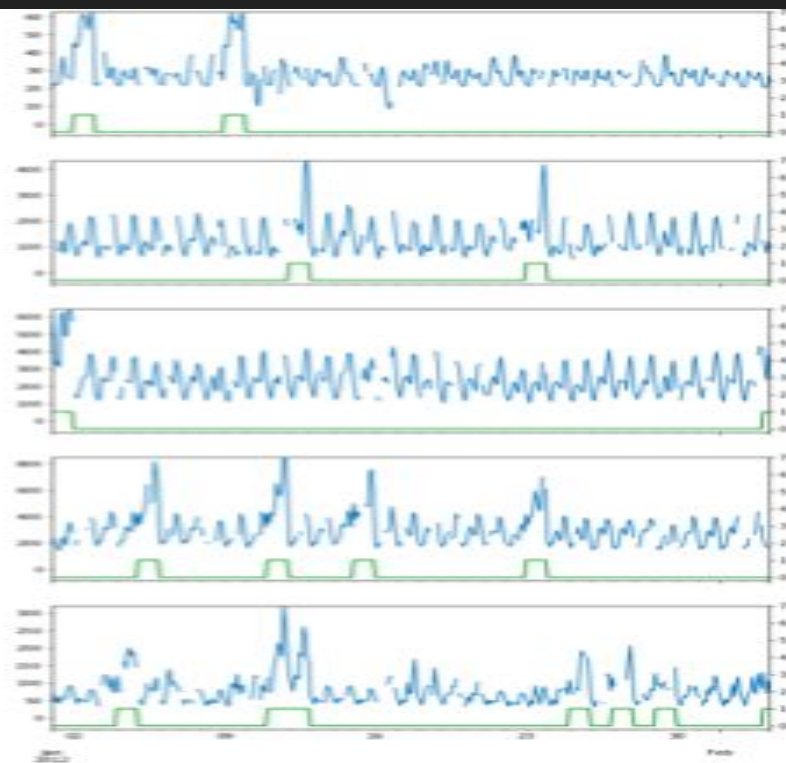
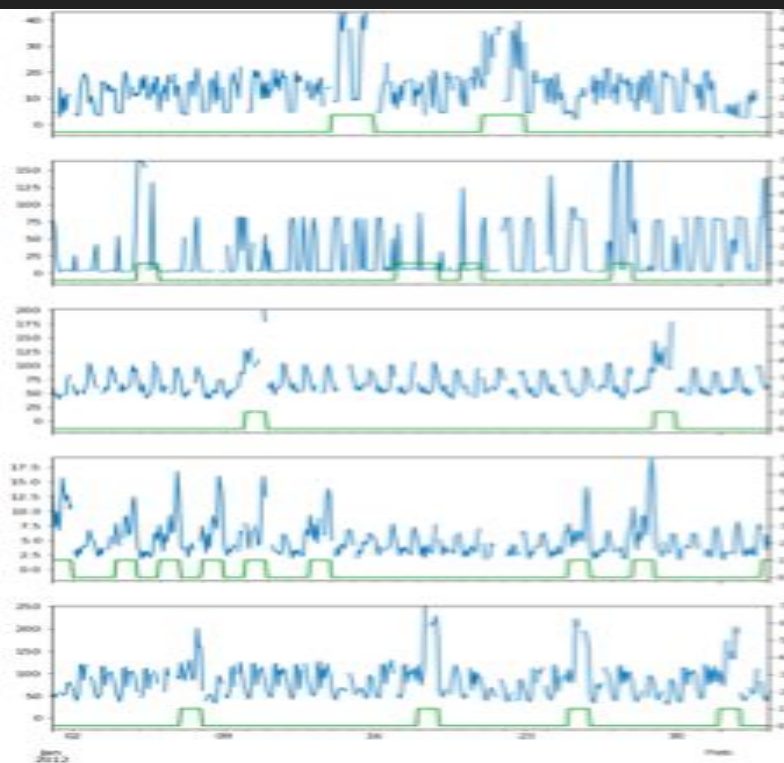
☒ show_samples

Run Interact

calling served model to generate predictions starting from 2014-11-16 00:00:00

```
/home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/ipykernel/__main__.py:19: FutureWarning: Addition/subtraction of integers and integer-arrays to Timestamp is deprecated, will be removed in a future version. Instead of adding/subtracting `n`, use `n * self.freq`  
/home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/ipykernel/__main__.py:46: FutureWarning: Creating a DatetimeIndex by passing range endpoints is deprecated. Use `pandas.date_range` instead.  
/home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/ipykernel/__main__.py:48: FutureWarning: Addition/subtraction of integers and integer-arrays to Timestamp is deprecated, will be removed in a future version. Instead of adding/subtracting `n`, use `n * self.freq`
```





Perform predictions by deploying it to an endpoint

