Evolution of Forecasting to Tackle Business Problems

From Standard Textbook Time Series Models to State of the Art Algorithms, Ensembling and Interpretability

SEGOLENE DESSERTINE-PANHARD AND YASH SHAH

Machine Learning Solutions Lab, Amazon Web Services, Inc.

Agenda

Agenda

Speaker Introductions

Why Forecast on the cloud?

History of Forecasting at Amazon

Some insights derived from our experience

Speaker Introductions

Segolene Dessertine-Panhard

Science Manager, AWS
Amazon Tenure (4 yrs)

PhD Sorbonne University,
Faculty at NYU FRE
working closely with **Peter**
on MMF (3 yrs)

Areas of Interest:
Supply Chain, Financial Modeling

Yash Shah

Science Manager, AWS
Amazon Tenure (5+ yrs)

MS Purdue University,
BE Mumbai University

Areas of Interest:
Healthcare, Supply Chain

Mission of Amazon Machine Learning Solutions Lab

Identify and **implement** our customers' highest-value ML use cases to **accelerate adoption.**

Some Customers we have worked with

ML FORECASTING SUCCESS BEING REALIZED ACROSS INDUSTRIES

more QUALITY 1st
Anaplan
FOXCONN PCE TECHNOLOGY DE JUÁREZ, S.A. DE C.V.
SWIGGY
RX | RETENTION X
PEAK
meesho
MOTOR
clearly
DEV FACTORY
CJ LOGISTICS
Heroleads
OMNYS
axiom telecom seriously unlike others
PSE PUGET SOUND ENERGY
DATUP
planalytics
AffordableTours.com
CASAONE

Why Forecast on the Cloud

Why Forecast on the Cloud

Data Across Systems

Data can come from various sources and needs consolidation into a single system

Easy to Use

fast prediction, easily parallelized if needed (on demand GPU and CPU)

Fully Managed

Automatically setting up pipeline, cleaning of resources and availability across regions

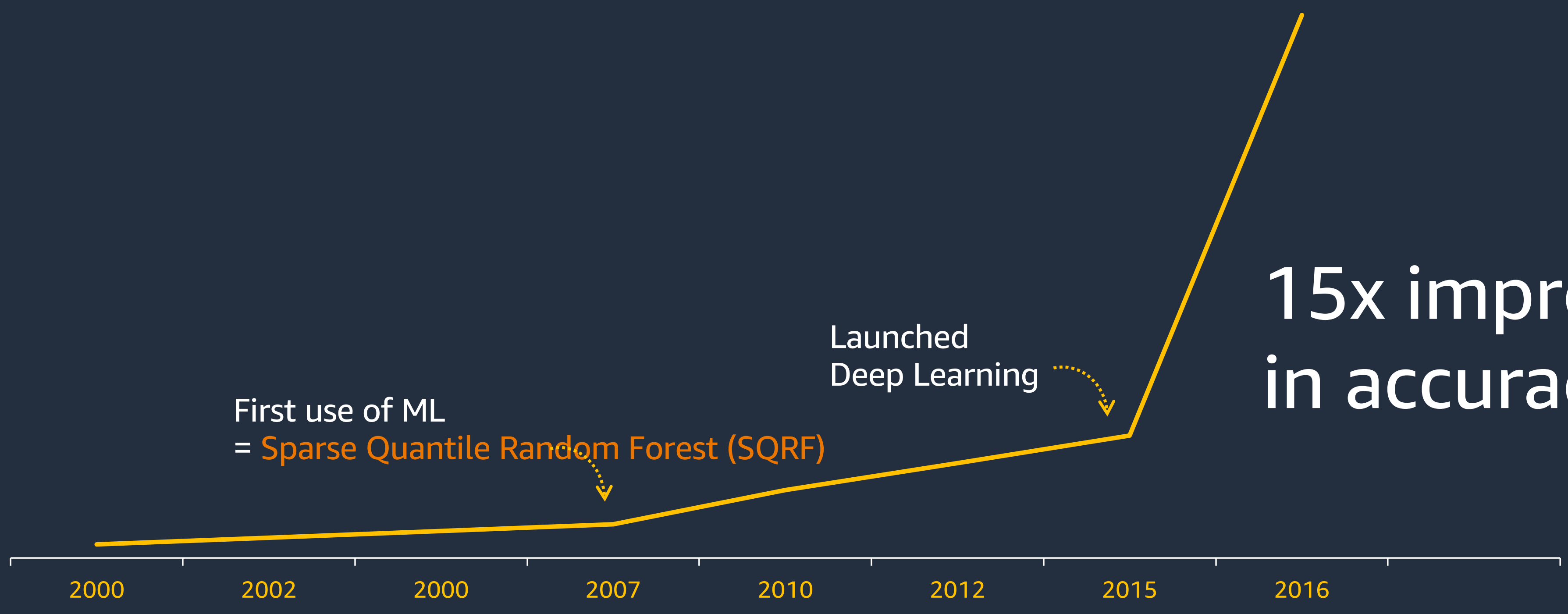
Improved Accuracy

Easy to try multiple models, model selection and evaluation is easier too

History of Forecasting at Amazon

Amazon Forecasting process' evolution

from local to global models
from statistical to deep learning forecasting models thanks to availability of large enough dataset



15x improvement in accuracy

Examples of SoTa models to address probabilistic modeling of large collections of TS

The goal of *forecasting* (Hyndman and Athanasopoulos, 2017) is to predict the probability distribution of future values $z_{i,T_{i+1}:T_{i+\tau}}$ given the past values $z_{i,1:T_i}$, the covariates $\mathbf{x}_{i,1:T_{i+\tau}}$, and the model parameters Φ:

$$p\left(z_{i,T_i+1:T_i+\tau} \mid z_{i,1:T_i}, \mathbf{x}_{i,1:T_i+\tau}; \Phi\right). \quad (1)$$

we follow mainly two modeling approaches: generative and discriminative depending on how the target Z is modeled

CATEGORY	MODELING
generative	$p(z_{i,1:T_i+\tau} \mid \mathbf{x}_{i,1:T_i+\tau}; \Phi)$
discriminative	$p(z_{i,T_i+1:T_i+\tau} \mid z_{i,1:T_i}, \mathbf{x}_{i,1:T_i+\tau}; \Phi)$

these models are available here : : https://github.com/awslabs/gluon-ts#available-models

(https://arxiv.org/pdf/1906.05264.pdf#cite.wavenet)

Examples of SoTa models to address probabilistic modeling of large collections of TS

- In a nutshell, **Generative models** assume that the given time series are generated from an unknown stochastic process p(Z|X; Φ) given the covariates X.

- The process is typically assumed to have some parametric structure with unknown parameters Φ. The unknown parameters of this stochastic process are typically estimated by maximizing the likelihood, which is the probability of the observed time series, {zi,1:Ti }, under the model p(Z|X; Φ), given the covariates {xi,1:Ti }.

- Once the parameters Φ are learned, the forecast distribution in Eq. (1) can be obtained from p(Z|X; Φ).

- Examples of generative models :
  - Several classical methods (ARIMA, ETS)
  - Gaussian processes
  - State space models
  - Deep State Space models (probabilistic time series forecasting approach that combines state space models with deep
  - learning)
  - Deep Factor models

Examples of SoTa models to address probabilistic modeling of large collections of TS

- **Discriminative models** model the conditional distribution (for a fixed τ) from Eq. (1) directly via a neural network. Compared to generative models, conditional models are more flexible, since they make less structural assumptions, and hence are also applicable to a broader class of application domains.

- We distinguish between auto-regressive and sequence-to-sequence models among discriminative model

- **Auto regressive models**:
  - Non-Parametric Time Series forecaster (NPTS)
  - DeepAR: auto-regressive RNN time series model which consists of a RNN (either using LSTM or GRU cells) that takes the previous time points and co-variates as input. DeepAR then either estimates parameters of a parametric distribution or a highly flexible parameterization of the quantile function.
  - Wavenet: auto-regressive neural network with dilated causal convolutions at its core (archetypical auto-regressive Convolutional Neural Network (CNN) models)

- **seq to seq models** : flexible sequence-to-sequence framework that makes it possible to combine generic encoder and decoder networks to create custom sequence-to-sequence models.
  - neural quantile regression models
  - Transformers

EVOLUTION OF FORECASTING TO TACKLE BUSINESS PROBLEMS

Name	Local/global	Architecture/method	Implementation	References
RForecast	Local	ARIMA, ETS, Croston, TBATS	Wrapped R package	paper
Prophet	Local	-	Wrapped Python package	paper
NaiveSeasonal	Local	-	Numpy	book section
Naive2	Local	-	Numpy	book section
NPTS	Local	-	Numpy	-
DeepAR	Global	RNN	MXNet, PyTorch	paper
SimpleFeedForward	Global	MLP	MXNet, PyTorch	-
DeepVAR	Global	RNN	MXNet	paper
GPVAR	Global	RNN, Gaussian process	MXNet	paper
LSTNet	Global	LSTM	MXNet	paper
DeepTPP	Global	RNN, temporal point process	MXNet	paper

aws

EVOLUTION OF FORECASTING TO TACKLE BUSINESS PROBLEMS

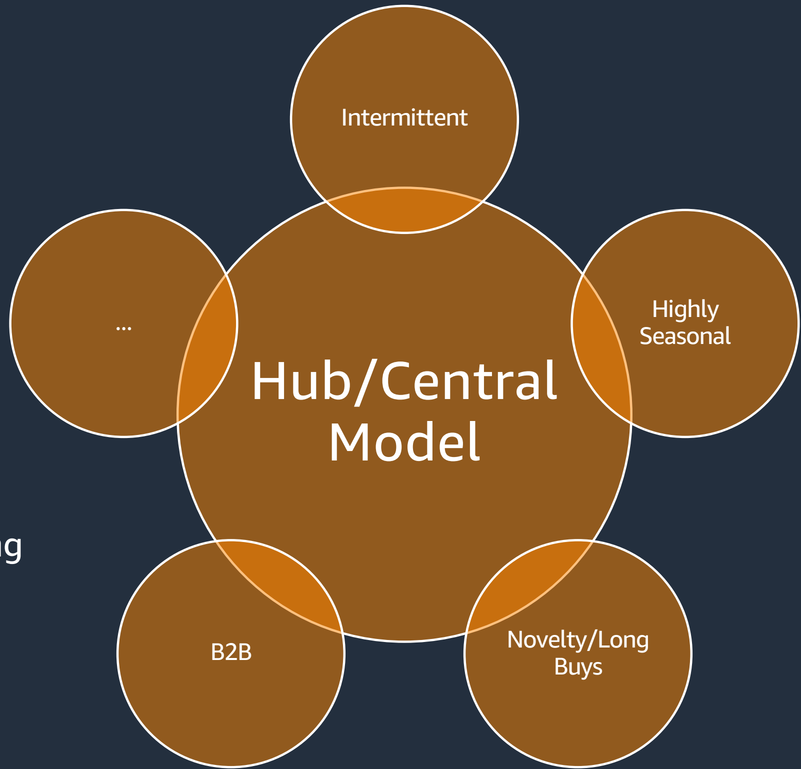
Name	Local/global	Architecture/method	Implementation	References
Deep Renewal Processes	Global	RNN	MXNet	paper
GPForecaster	Global	MLP, Gaussian process	MXNet	-
MQ-CNN	Global	CNN encoder, MLP decoder	MXNet	paper
MQ-RNN	Global	RNN encoder, MLP encoder	MXNet	paper
N-BEATS	Global	MLP, residual links	MXNet	paper
Rotbaum	Global	XGBoost, Quantile Regression Forests, LightGBM, Level Set Forecaster	Numpy	paper
Causal Convolutional Transformer	Global	Causal convolution, self attention	MXNet	paper
Temporal Fusion Transformer	Global	LSTM, self attention	MXNet	paper
Transformer	Global	MLP, multi-head attention	MXNet	paper
WaveNet	Global	Dilated convolution	MXNet	paper
DeepState	Global	RNN, state-space model	MXNet	paper
DeepFactor	Global	RNN, state-space model, Gaussian process	MXNet	paper

aws

Some insights derived from our experience

One-Size Fits All?

- A ‘hub’ model or model ensembles that solves for the majority of the use cases and special use cases are solved via ‘spoke’ models.
- We then continuously iterate on a spoke, generalizing it, and integrate it back to hub.
- We need to decide on which use cases to tackle by prioritizing (eg. ABC)
- No need to demolish and build. Lift and replace.
- Have a shadow pipeline for live estimation.
- Different use cases might need different approaches for forecasting and assessment



Metrics we typically Use

Underbias: It measures how far off forecast is from demand in direction that forecast is smaller than demand, as a percentage of demand.

$$UB(_p) = \max\{0, d - qp\} \;/\; d$$

Overbias: It measures how far off forecast is from demand in direction that forecast is greater than demand, as a percentage of demand.

$$OB(_p) = \max(0, qp - d) \;/\; d$$

Quantile Loss: measures combines under bias and over bias.

$$quantileLoss(_p) = (p * UB(_p) + (1 - p) * OB(_p))$$

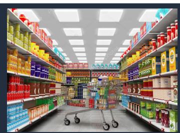
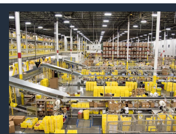
Calibration: measures the probability that actual demand is below some quantile point. A calibrated forecast is one for which the outcomes predicted to occur with probability (p) actually occur (p) of the time

$$calibration(p) = \frac{1}{nbProduct} \sum_{Product} 1[d \leq p]$$

aws

The case for Forecasting

Impact of under and over forecasting

	 Inventory planning	 Workforce planning	 Capacity planning	 Financial planning
Over-forecasting	Excess inventory	Unutilized labor	Uncapitalized infrastructure	Depleted cash reserves
Under-forecasting	Lost sale	Overtime costs	Unmet demand	Undercutting

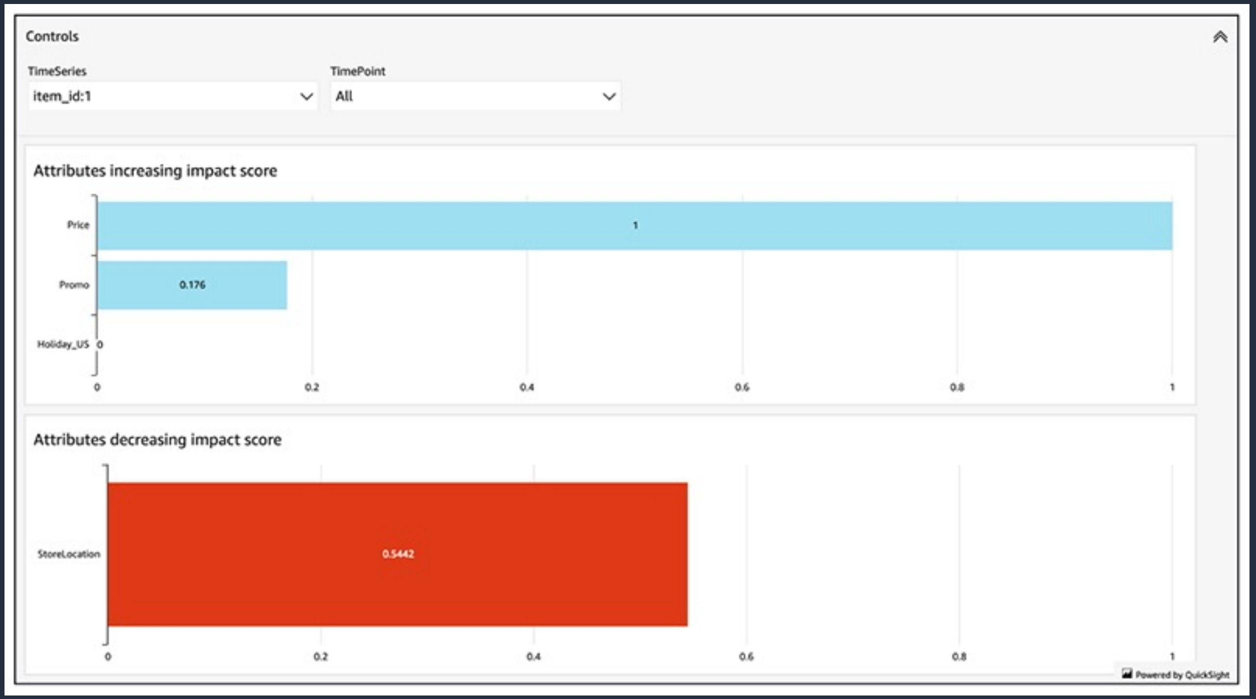
Examples of metric selection strategies

Use case	wQL	Avg.WQL	WAPE	RMSE	MAPE	MASE
Optimizing for under-forecasting or over-forecasting, which may have different implications	X	X				
Prioritizing popular items or items with high demand is more important than low-demand items	X	X	X			
Emphasizing business costs related to large deviations in forecasts errors				X	X	
Assessing sparse datasets with 0 demand for most items and historical data points	X	X	X			
Measuring seasonality impacts						X

What after Forecasts are generated?

Interpretability

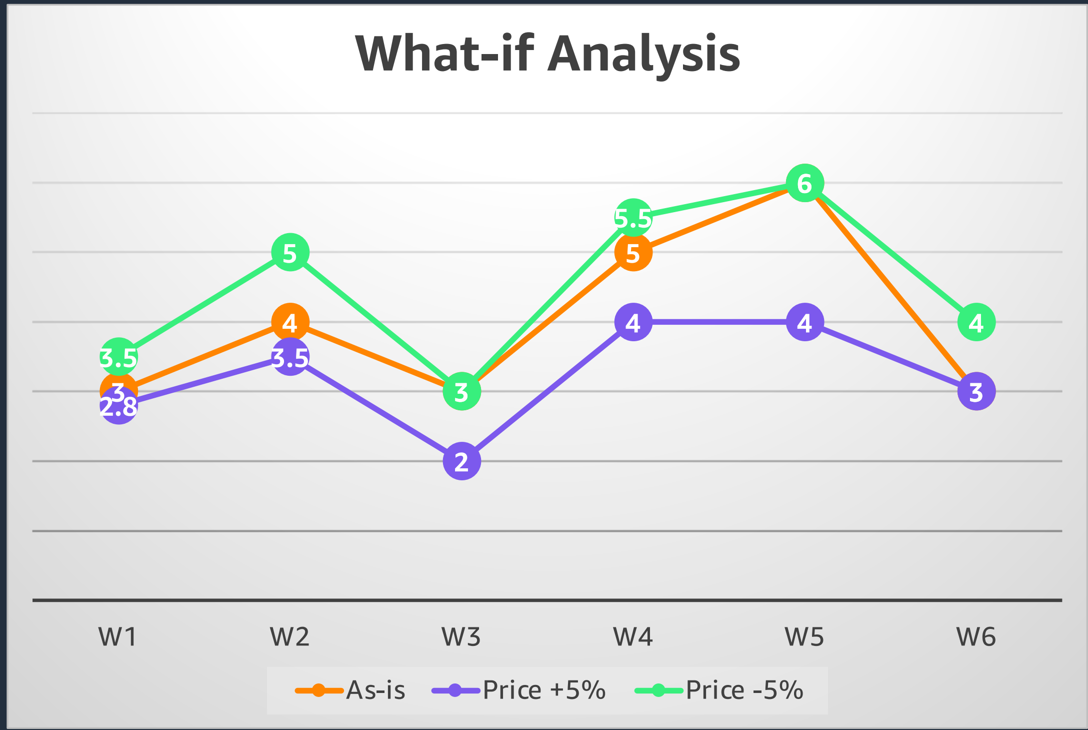
- Increased chance of adoption and success
- Understand the strengths and weaknesses of a model
- Learn and discover new insights from data



What after Forecasts are generated?

What if Analyses

- Create contingency plans that management can utilize under certain circumstances
- Easier decision making
- Simulate and assess impact of decisions on direct and indirect users
- Create alarms and triggers



What after Forecasts are generated?

Expert in the Loop

- Expert might have additional information about future application requirements like marketing campaigns that the forecast calculation is unable to take into account
- Sometimes it is to adjust before and after for specific events into the forecast like holiday seasons and unknown macro economic events like COVID-19

Plan Outputs

Metric	Jan 6 - Jan 12, 2022	Jan 13
Forecasting Inputs		
Forecasted Contact Volume	6761	
Forecasted Average Handling Time (AHT), seconds	917	
Outputs	Original value: 79	
Required FTEs (without Shrinkage)	90	
Forecasted Occupancy %	50%	

Thank you!

Segolene Dessertine-Panhard
desserti@amazon.com

Yash Shah
syash@amazon.com