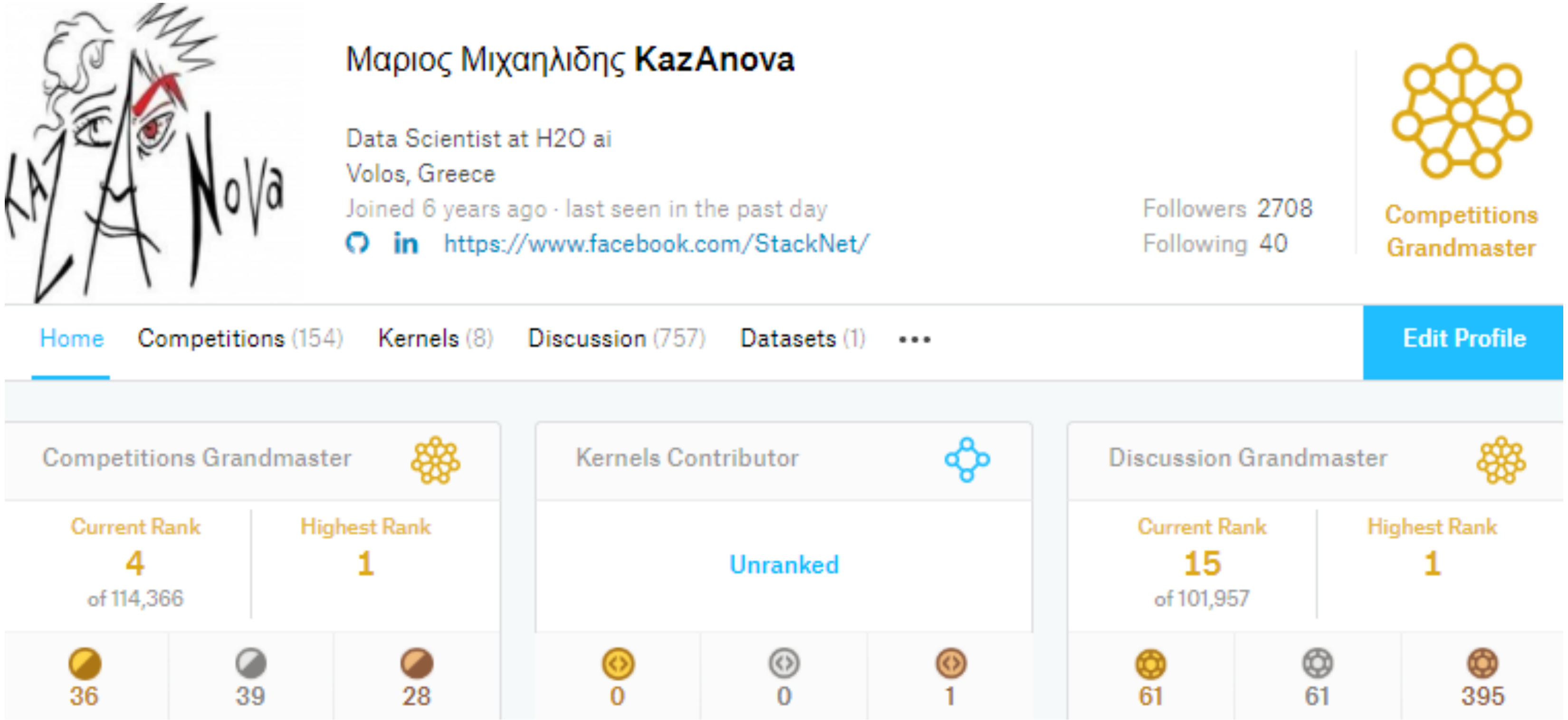
Background

- Competitive data scientist at **H2O**.ai
- PhD in ensemble methods at UCL
- Former kaggle #1 – over 150+ competitions



H2O.ai Product Suite

In-memory, distributed machine learning algorithms with H2O Flow GUI

H2O AI open source engine integration with Spark

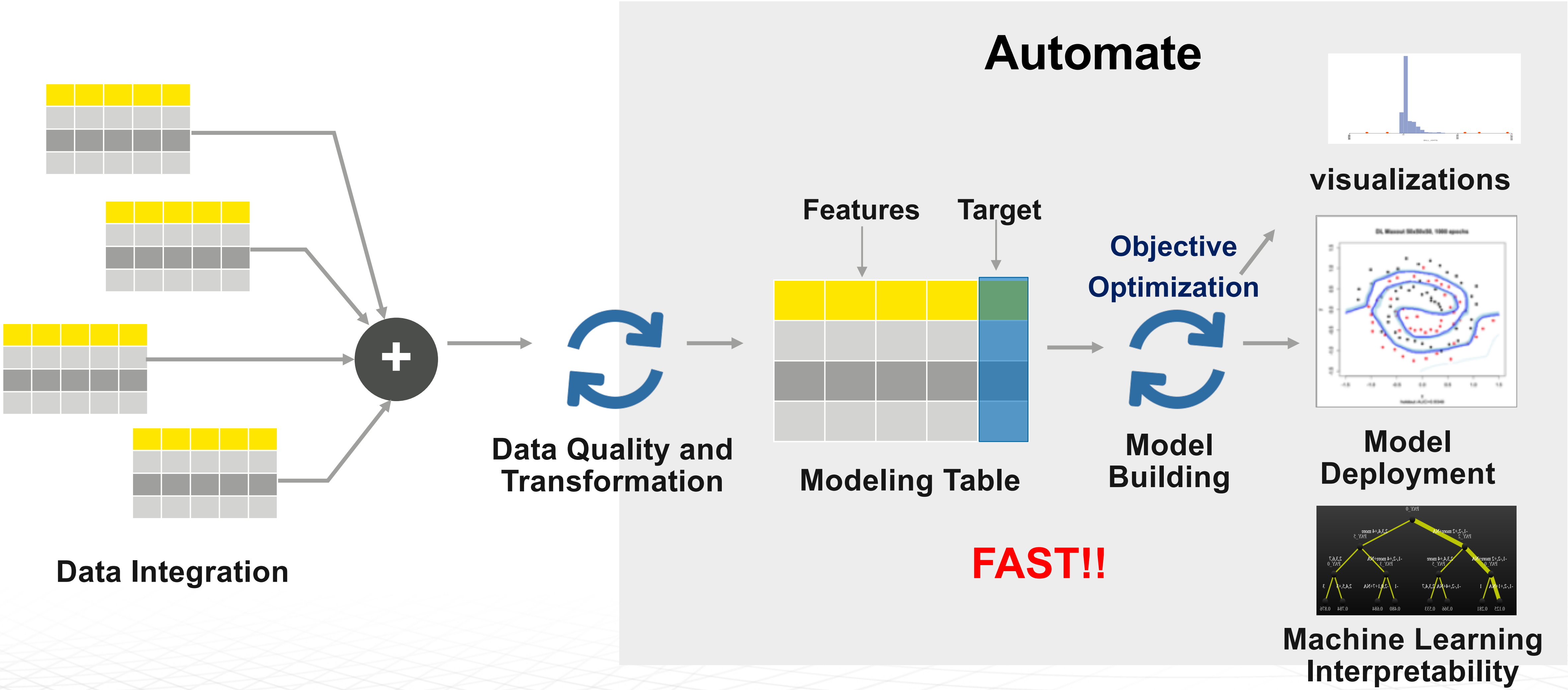
Lightning fast machine learning on GPUs

Automatic feature engineering, machine learning and interpretability

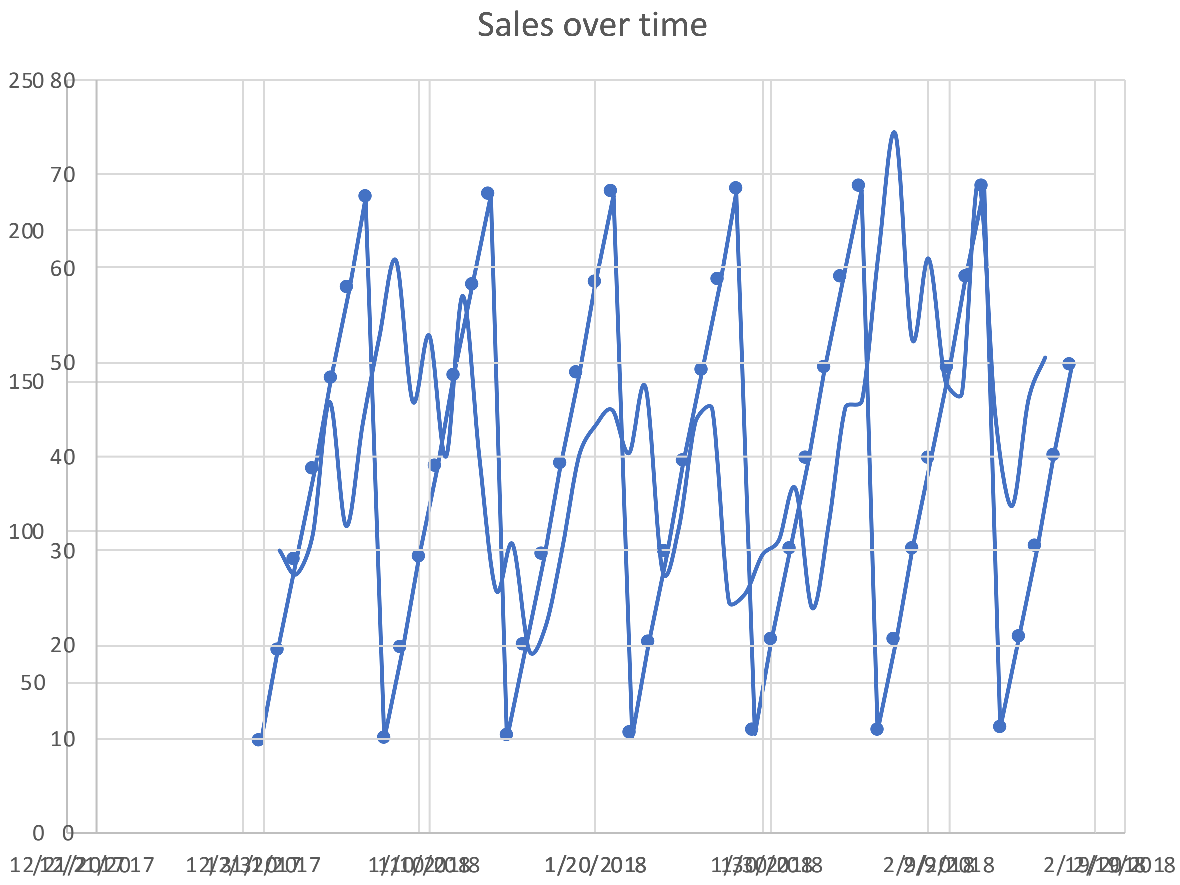
- 100% open source – Apache V2 licensed
- Built for data scientists – interface using R, Python or H2O Flow (interactive notebook interface)
- Enterprise support subscriptions

- Fully automated machine learning from ingest to deployment
- User licenses on a per seat basis (annual subscription)

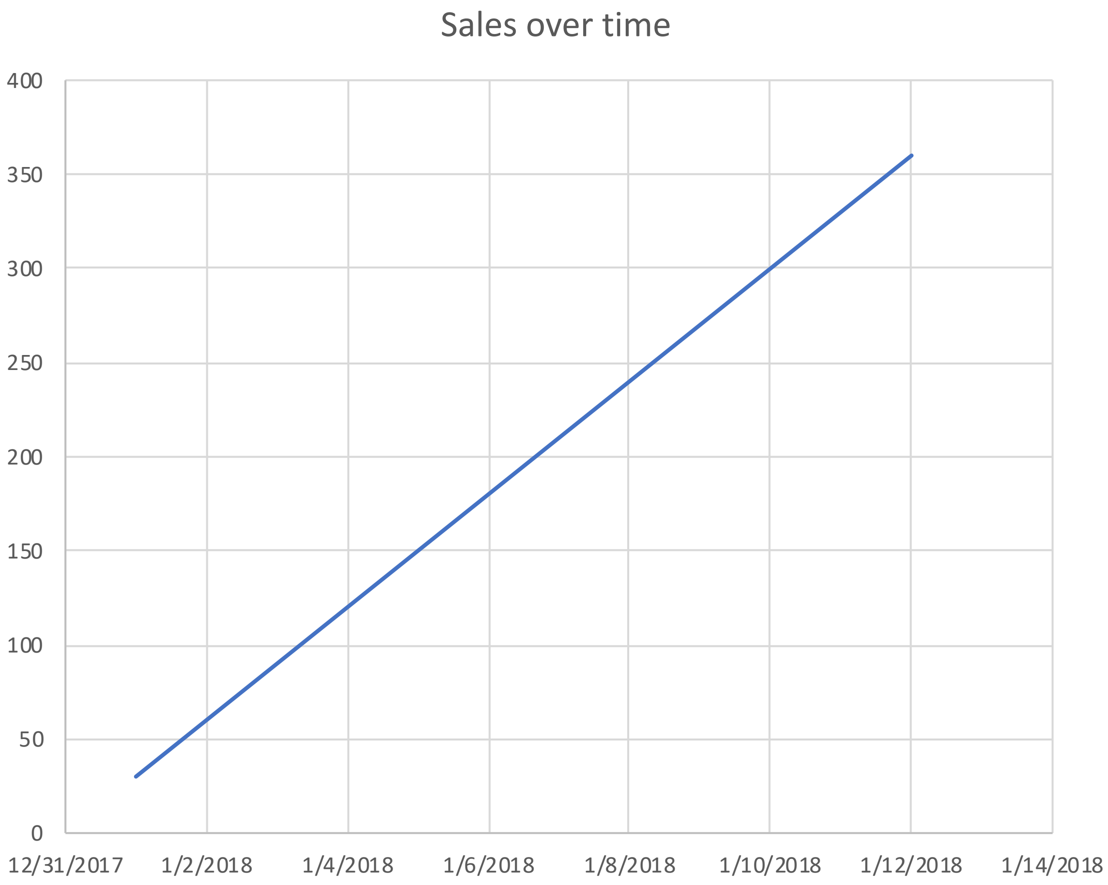
Driverless AI Workflow



What is a Time Series Problem?

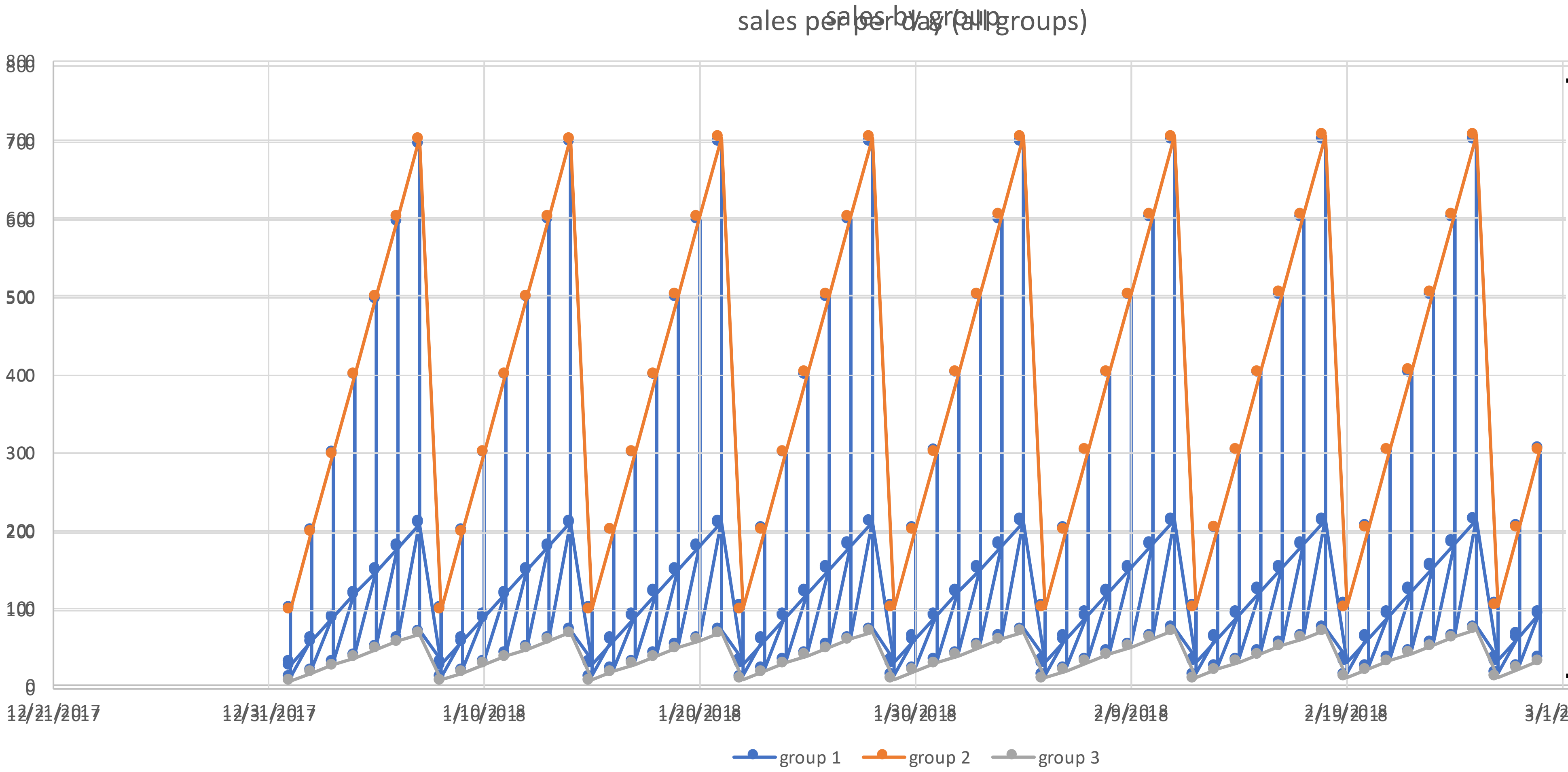


Nonlinear (seasonal) relationship



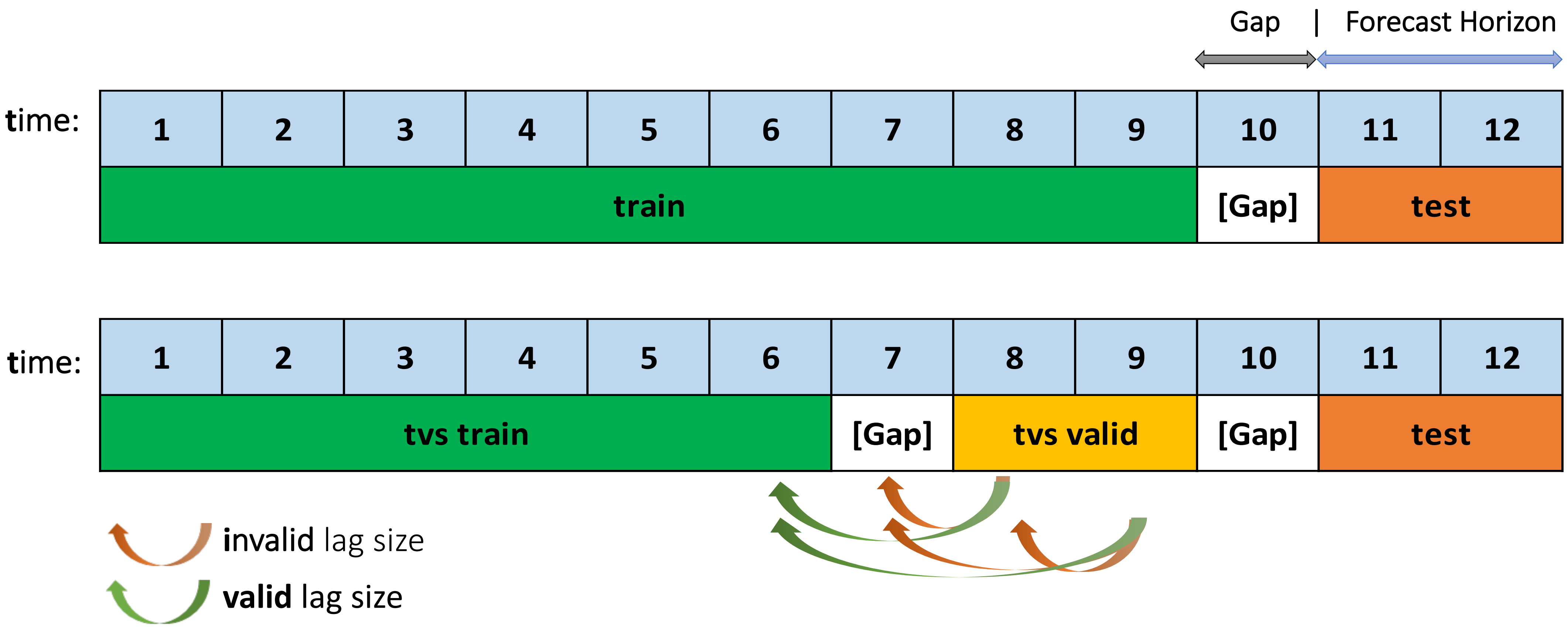
Linear relationship

Time Groups



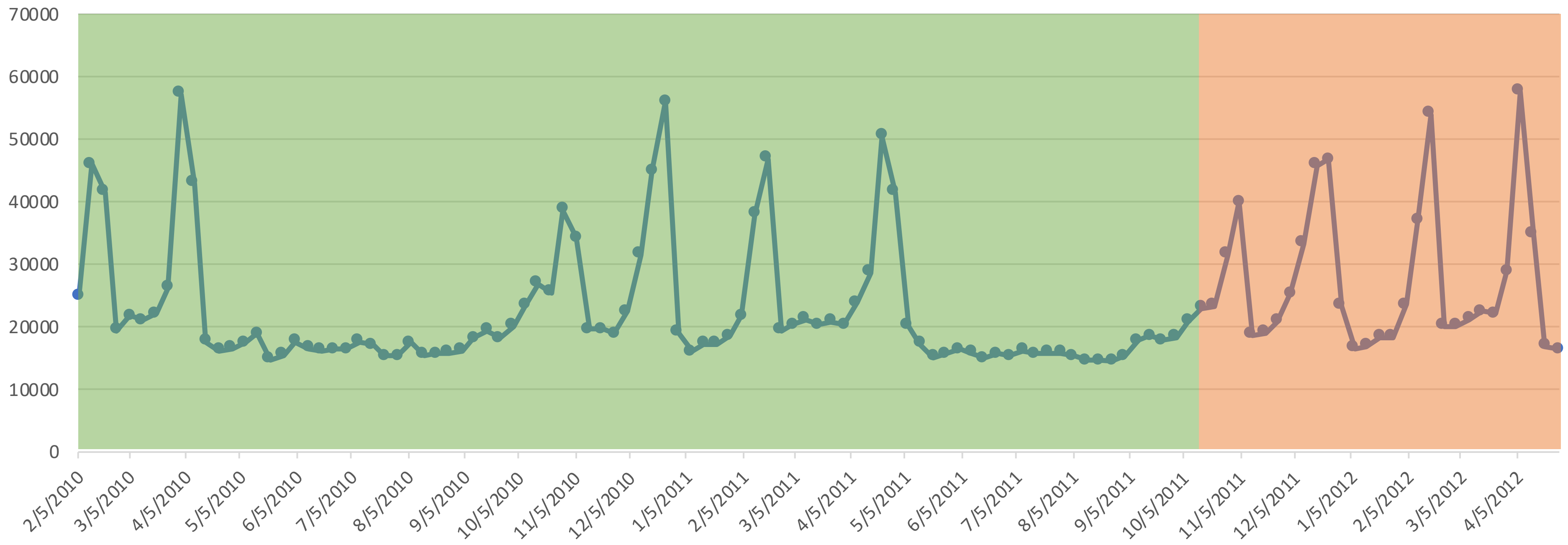
time	groups	sales
01/01/2018	group1	30
01/01/2018	group2	100
01/01/2018	group3	10
02/01/2018	group1	60.2
02/01/2018	group2	200.2
02/01/2018	group3	20.2
03/01/2018	group1	90.3
03/01/2018	group2	300.3
03/01/2018	group3	30.3
04/01/2018	group1	120.4
04/01/2018	group2	400.4
04/01/2018	group3	40.4

Modeling Foundation



Validation Schemas #1

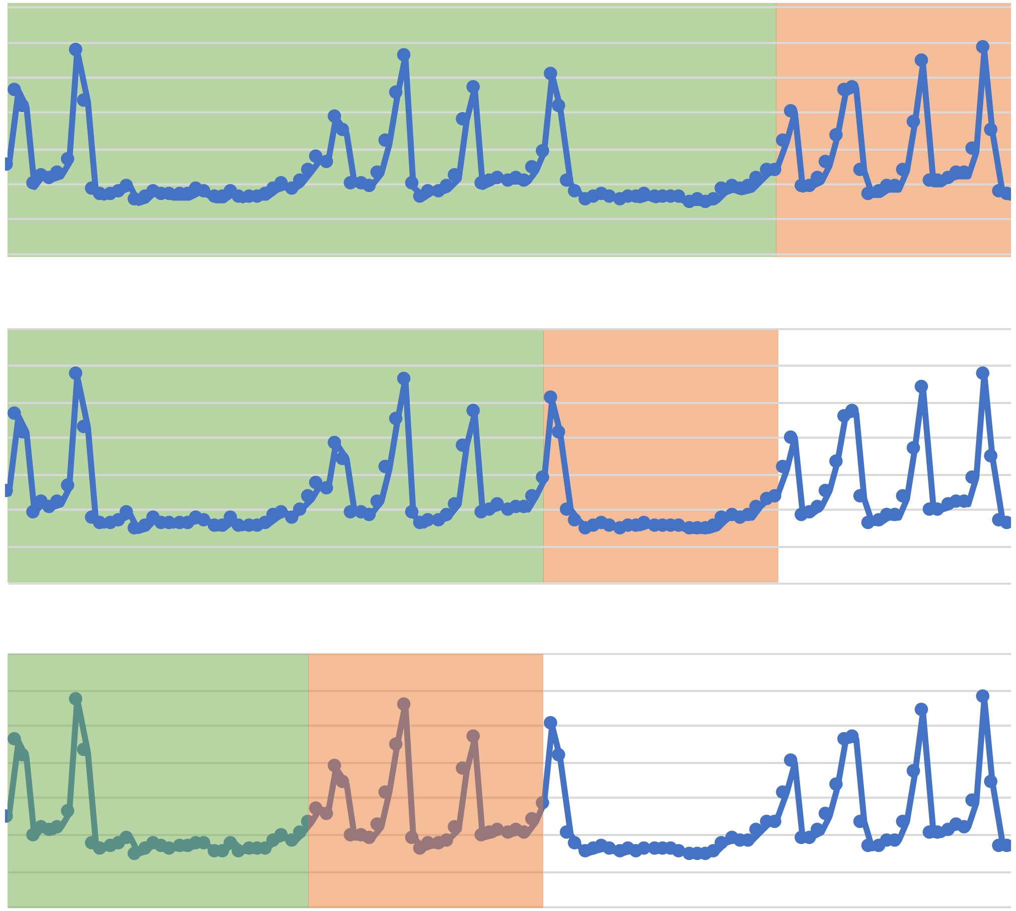
- Single time split (most recent training data becomes validation)



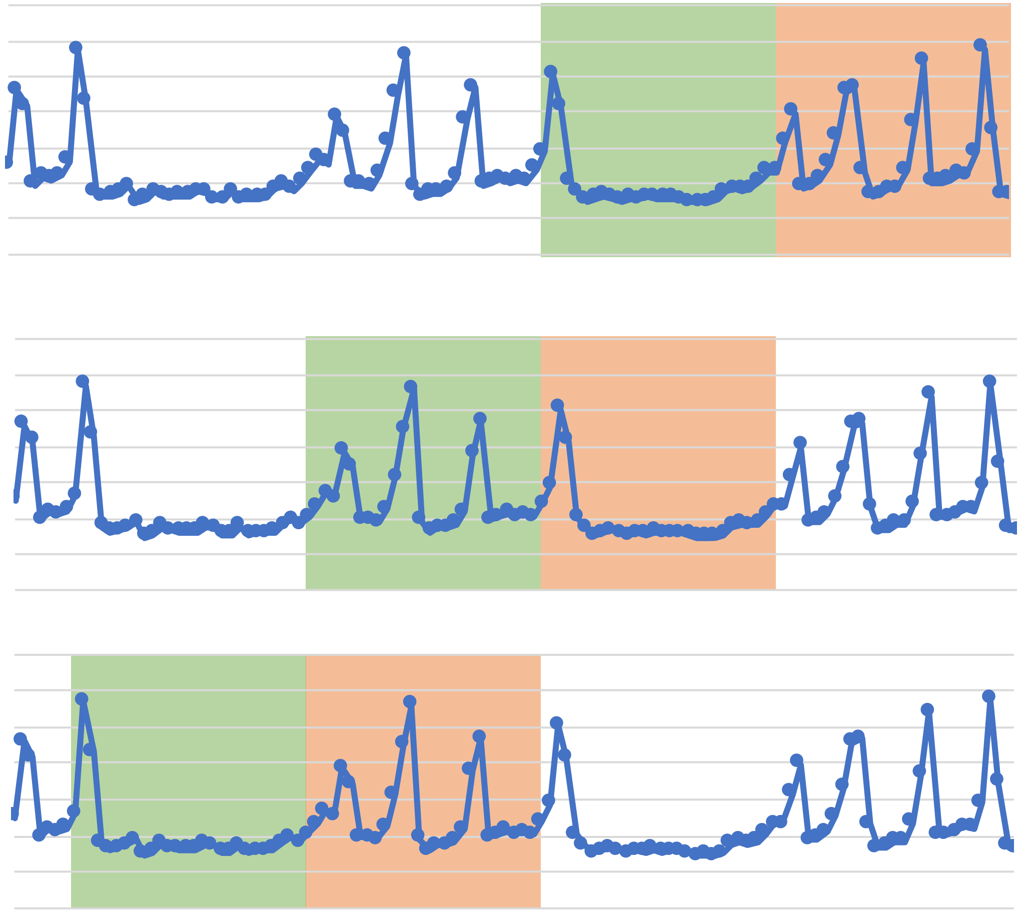
Validation Schemas #2

- Multi window validation

Rolling window with **adjusting** training size



Rolling window with **constant** training size



Feature Engineering: Decomposing the date

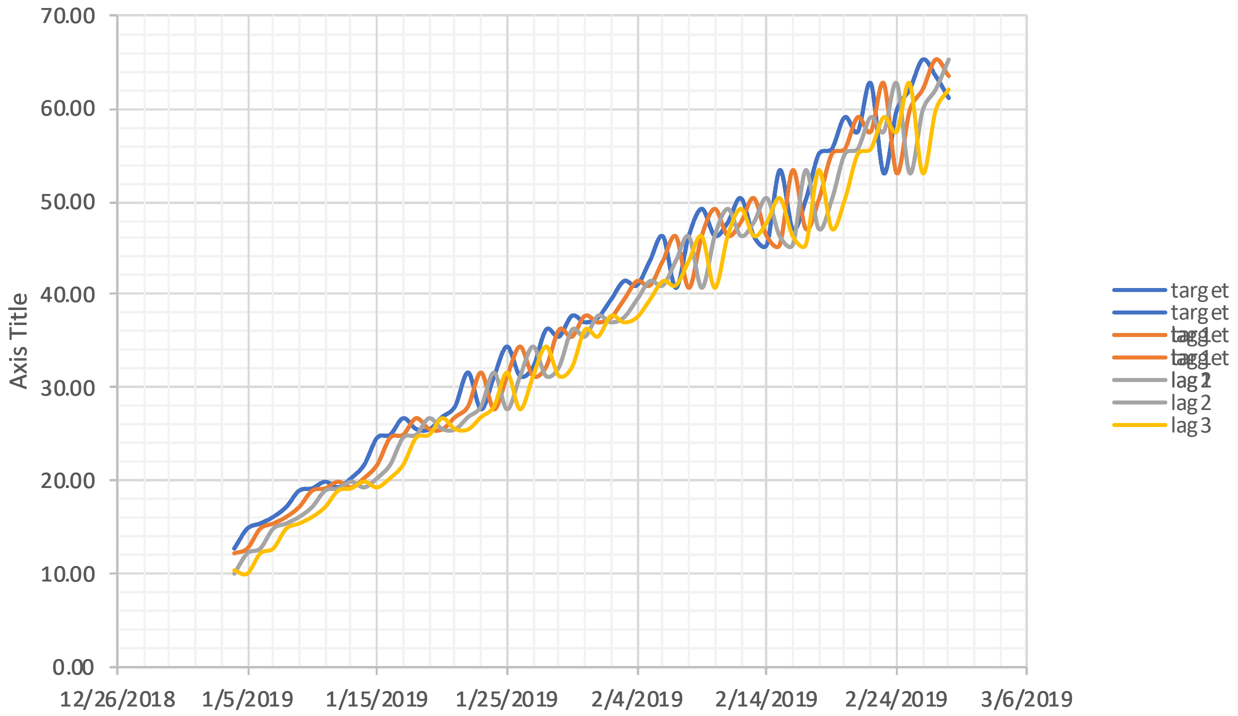
Date
1/1/2018
2/1/2018
3/1/2018
4/1/2018
5/1/2018
6/1/2018
7/1/2018
8/1/2018
9/1/2018
10/1/2018

Day	Month	Year	Weekday	Weeknum	IsHoliday
1	1	2018	2	1	1
2	1	2018	3	1	0
3	1	2018	4	1	0
4	1	2018	5	1	0
5	1	2018	6	1	0
6	1	2018	7	1	0
7	1	2018	1	2	0
8	1	2018	2	2	0
9	1	2018	3	2	0
10	1	2018	4	2	0

Feature Engineering: Lags

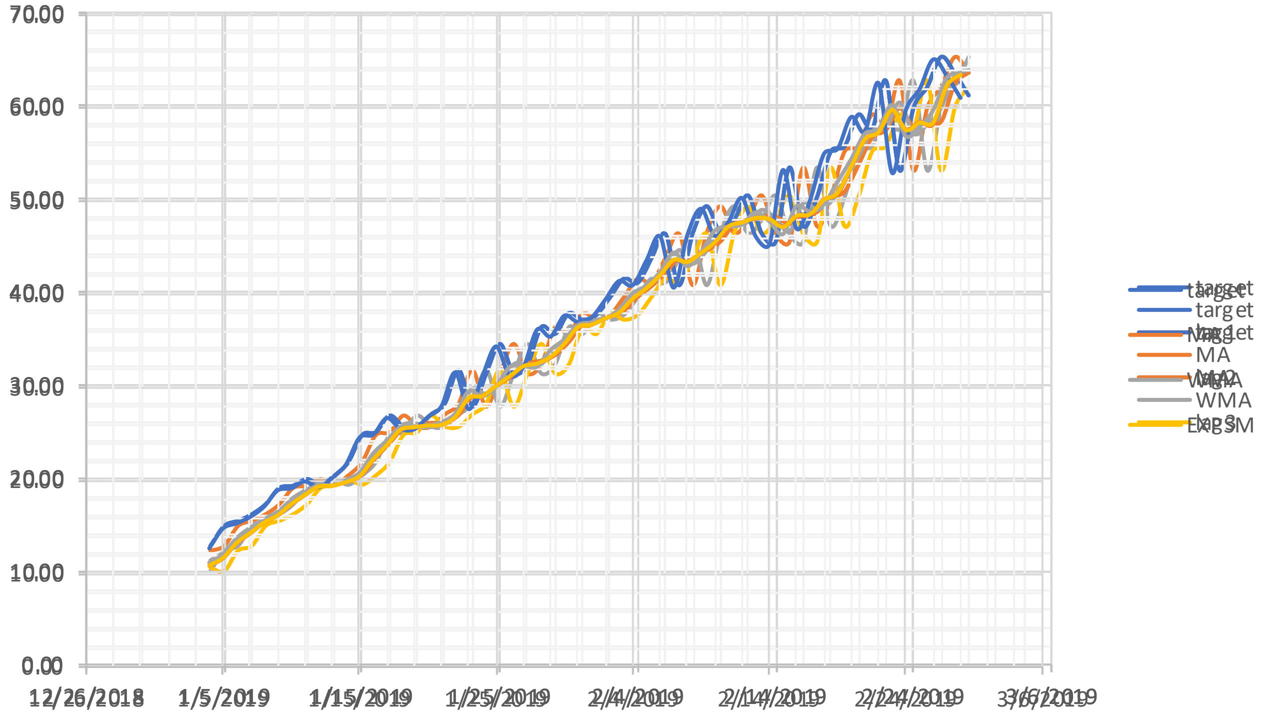
date	target
01/01/2019	10.40
02/01/2019	10.04
03/01/2019	12.22
04/01/2019	12.74
05/01/2019	14.87
06/01/2019	15.43
07/01/2019	16.13
08/01/2019	17.20
09/01/2019	18.96
10/01/2019	19.20
11/01/2019	19.92
12/01/2019	19.31
13/01/2019	20.30
14/01/2019	21.73
15/01/2019	24.64

lag1	lag2	lag3
10.40		
10.04	10.40	
12.22	10.04	10.40
12.74	12.22	10.04
14.87	12.74	12.22
15.43	14.87	12.74
16.13	15.43	14.87
17.20	16.13	15.43
18.96	17.20	16.13
19.20	18.96	17.20
19.92	19.20	18.96
19.31	19.92	19.20
20.30	19.31	19.92
21.73	20.30	19.31



Feature Engineering: Windows

date	target	lag1	lag2	lag3	STD	MAX	SKEW
01/01/2019	10.40						
02/01/2019	10.04	10.40					
03/01/2019	12.22	10.04	10.40				
04/01/2019	12.74	12.22	10.04	10.40	[illegible]	[illegible]	[illegible]
05/01/2019	14.87	12.74	12.22	10.04	[illegible]	[illegible]	[illegible]
06/01/2019	15.43	14.87	12.74	12.22	[illegible]	[illegible]	[illegible]
07/01/2019	16.13	15.43	14.87	12.74	[illegible]	[illegible]	[illegible]
08/01/2019	17.20	16.13	15.43	14.87	[illegible]	[illegible]	[illegible]
09/01/2019	18.96	17.20	16.13	15.43	[illegible]	[illegible]	[illegible]
10/01/2019	19.20	18.96	17.20	16.13	[illegible]	[illegible]	[illegible]
11/01/2019	19.92	19.20	18.96	17.20	[illegible]	[illegible]	[illegible]
12/01/2019	19.31	19.92	19.20	18.96	[illegible]	[illegible]	[illegible]
13/01/2019	20.30	19.31	19.92	19.20	[illegible]	[illegible]	[illegible]
14/01/2019	21.73	20.30	19.31	19.92	[illegible]	[illegible]	[illegible]
15/01/2019	24.64	21.73	20.30	19.31	[illegible]	[illegible]	[illegible]



For hyper parameter **a=0.95**

12.22 x 3 + 10.04 x 2 + 10.40 x 1 / (3+2+1) = **10.88**

12.22 + 10.04 + 10.40 / 3 = **11.18**

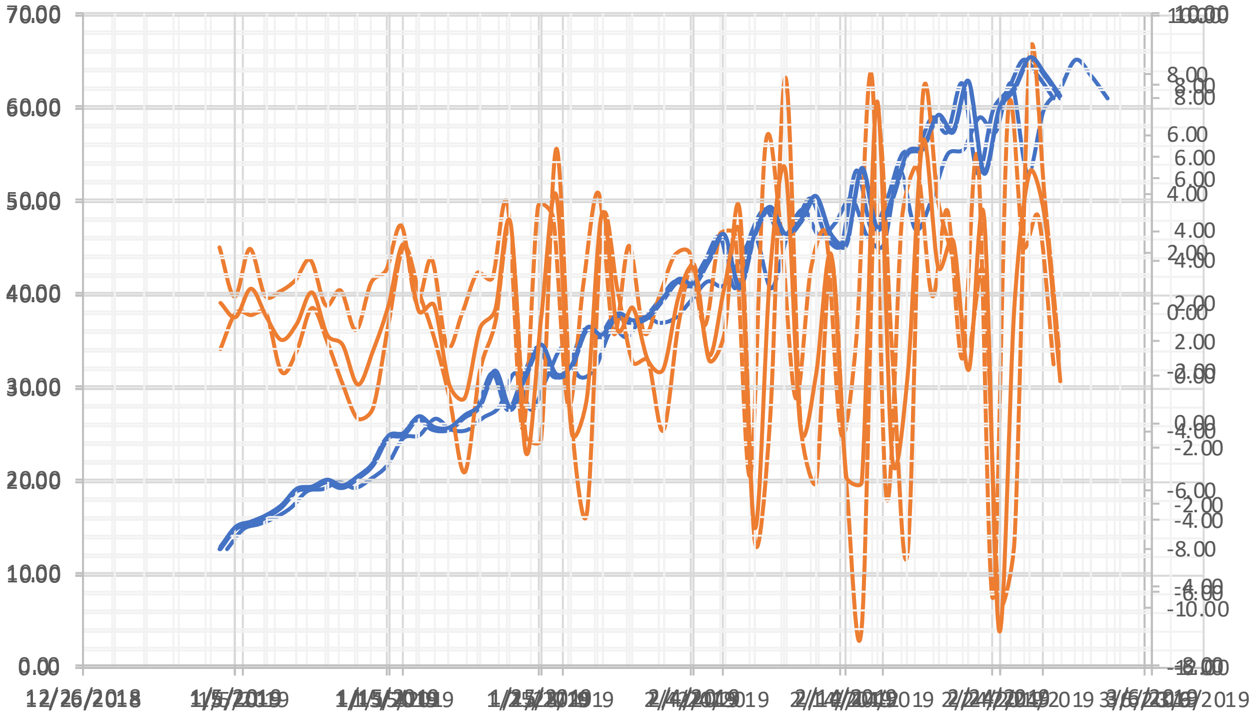
(12.22 x (0.95**1) + 10.04 x (0.95**2) +10.40 x (0.95**3)) /((0.95**1) + (0.95**2) + (0.95**3)) = **10.92**

Other descriptive statistics

Max, min, median, std, kurtosis, skewness

Feature Engineering: Interractions

date	target	lag1	lag2	lag3	diff1	diff2	MAdif	div1
01/01/2019	10.40							
02/01/2019	10.04	10.40						
03/01/2019	12.22	10.04	10.40					
04/01/2019	12.74	12.22	10.04	10.40	2.18	2.70	2.44	1.22
05/01/2019	14.87	12.74	12.22	10.04	0.52	2.65	1.59	1.04
06/01/2019	15.43	14.87	12.74	12.22	2.13	2.69	2.41	1.17
07/01/2019	16.13	15.43	14.87	12.74	0.56	1.26	0.91	1.04
08/01/2019	17.20	16.13	15.43	14.87	0.70	1.77	1.24	1.05
09/01/2019	18.96	17.20	16.13	15.43	1.07	2.83	1.95	1.07
10/01/2019	19.20	18.96	17.20	16.13	1.75	1.99	1.87	1.10
11/01/2019	19.92	19.20	18.96	17.20	0.24	0.96	0.60	1.01
12/01/2019	19.31	19.92	19.20	18.96	0.72	0.11	0.42	1.04
13/01/2019	20.30	19.31	19.92	19.20	-0.61	0.38	-0.12	0.97
14/01/2019	21.73	20.30	19.31	19.92	1.00	2.42	1.71	1.05
15/01/2019	24.64	21.73	20.30	19.31	1.43	4.33	2.88	1.07



Diff1=lag1-lag2

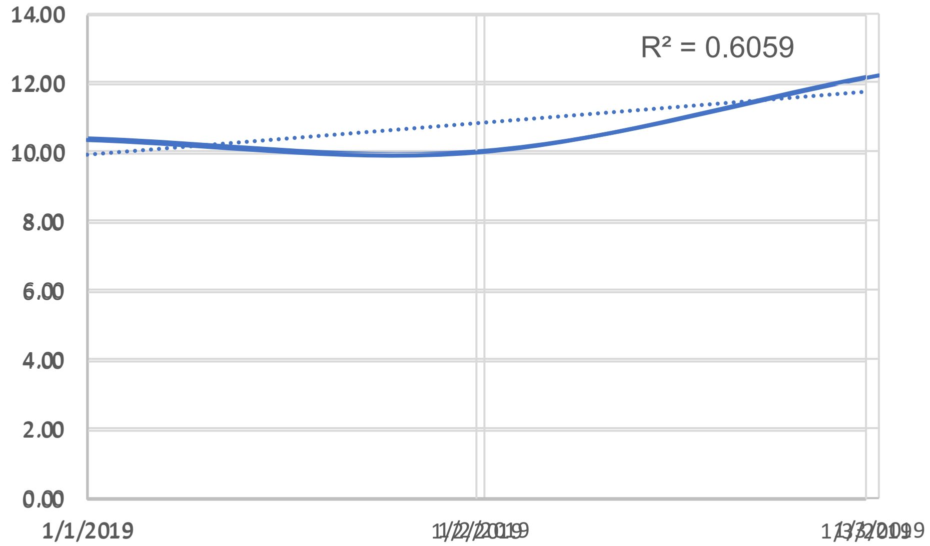
Diff2=lag1-lag3

MAdiff= (Diff1+ Diff2)/2

Div1=lag1/lag2

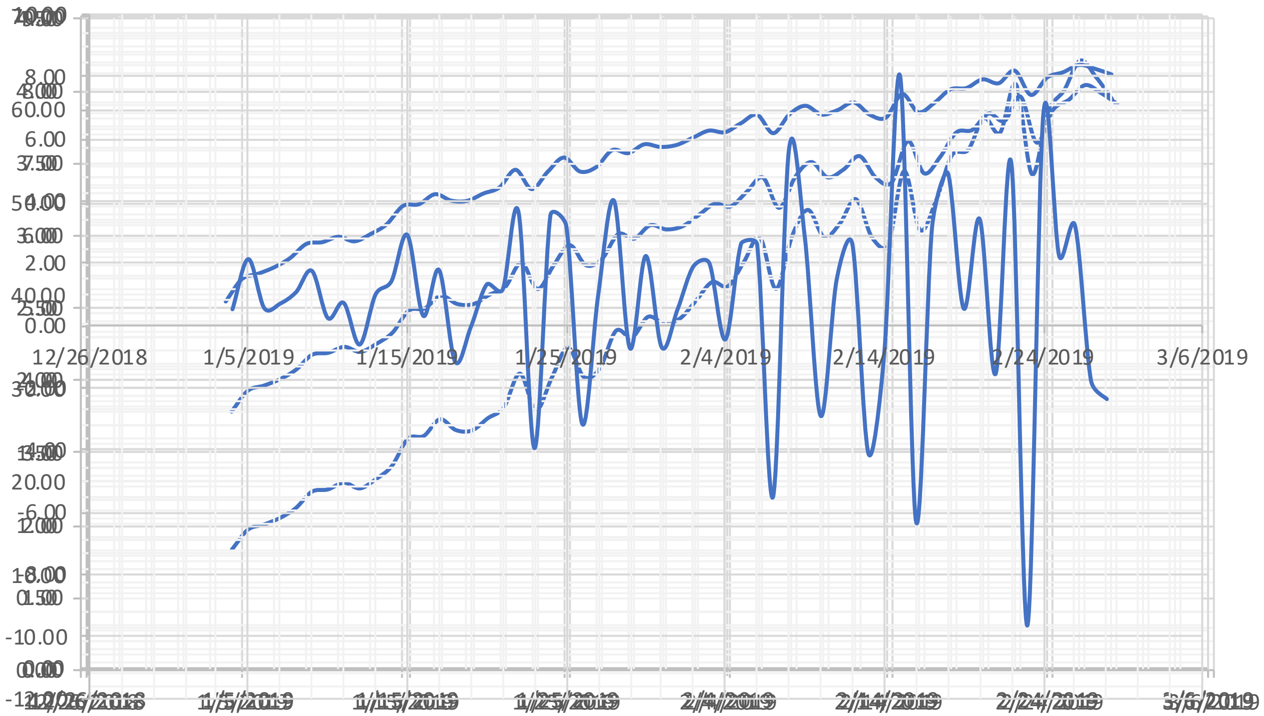
Feature Engineering: trends

date	target	lag1	lag2	lag3	correl
01/01/2019	10.40				
02/01/2019	10.04	10.40			
03/01/2019	12.22	10.04	10.40		
04/01/2019	12.74	12.22	10.04	10.40	0.78
05/01/2019	14.87	12.74	12.22	10.04	0.94
06/01/2019	15.43	14.87	12.74	12.22	0.94
07/01/2019	16.13	15.43	14.87	12.74	0.95
08/01/2019	17.20	16.13	15.43	14.87	1.00
09/01/2019	18.96	17.20	16.13	15.43	0.99
10/01/2019	19.20	18.96	17.20	16.13	0.99
11/01/2019	19.92	19.20	18.96	17.20	0.92
12/01/2019	19.31	19.92	19.20	18.96	0.96
13/01/2019	20.30	19.31	19.92	19.20	0.14
14/01/2019	21.73	20.30	19.31	19.92	0.38
15/01/2019	24.64	21.73	20.30	19.31	0.99



Feature Engineering: Target Transformations

date	target	sqrt	log	differ
01/01/2019	10.40	3.22	2.34	
02/01/2019	10.04	3.17	2.31	-0.36
03/01/2019	12.22	3.50	2.50	2.18
04/01/2019	12.74	3.57	2.54	0.52
05/01/2019	14.87	3.86	2.70	2.13
06/01/2019	15.43	3.93	2.74	0.56
07/01/2019	16.13	4.02	2.78	0.70
08/01/2019	17.20	4.15	2.85	1.07
09/01/2019	18.96	4.35	2.94	1.75
10/01/2019	19.20	4.38	2.95	0.24
11/01/2019	19.92	4.46	2.99	0.72
12/01/2019	19.31	4.39	2.96	-0.61
13/01/2019	20.30	4.51	3.01	1.00
14/01/2019	21.73	4.66	3.08	1.43
15/01/2019	24.64	4.96	3.20	2.91



Candidates for Lag-Sizes

- Ranking based on autocorrelation

- Pre-defined intervals (based on estimated frequency)

  Daily data

  - [7, 14, 21, …]
  - [14, 28, 32, …]
  - …

  Weekly data

  - [2, 4, 6, 8, …]
  - [4, 8, 12, 16, …]
  - …

  …

Regularization of Lag-Features

- Dropouts
  - Random replacement of actual lag-values by „n.a.“
  - Align frequency of available lag information between train and validation/test

Using top-performing Algorithms

Microsoft LightGBM

dmlc XGBoost

scikit learn

H2O

K + TensorFlow

PYTORCH

SciPy

data.table

NumPy

Pandas

Genetic algorithm approach

Date	x1	x2	x3	x4	y
01/01/2019	200	cust1	0.01	prod1	32
02/01/2019	250	cust1	0.45	prod2	21
03/01/2019	50	cust1	0.51	prod3	20
01/01/2019	45	cust2	0.79	prod1	18
02/01/2019	125	cust2	0.72	prod2	27
03/01/2019	400	cust2	0.28	prod3	35
01/01/2019	230	cust3	0.68	prod1	37
02/01/2019	210	cust3	0.35	prod2	30
03/01/2019	500	cust3	0.28	prod3	28
01/01/2019	505	cust4	0.63	prod1	29
02/01/2019	150	cust4	0.53	prod2	40
03/01/2019	170	cust4	0.33	prod3	35

Iteration 1/10

Iteration 2/10

+ **Random Transformation**

XGBoost $\underbrace{\quad\quad}_{\text{x2 + x4}}$

lag1(x3,x4)

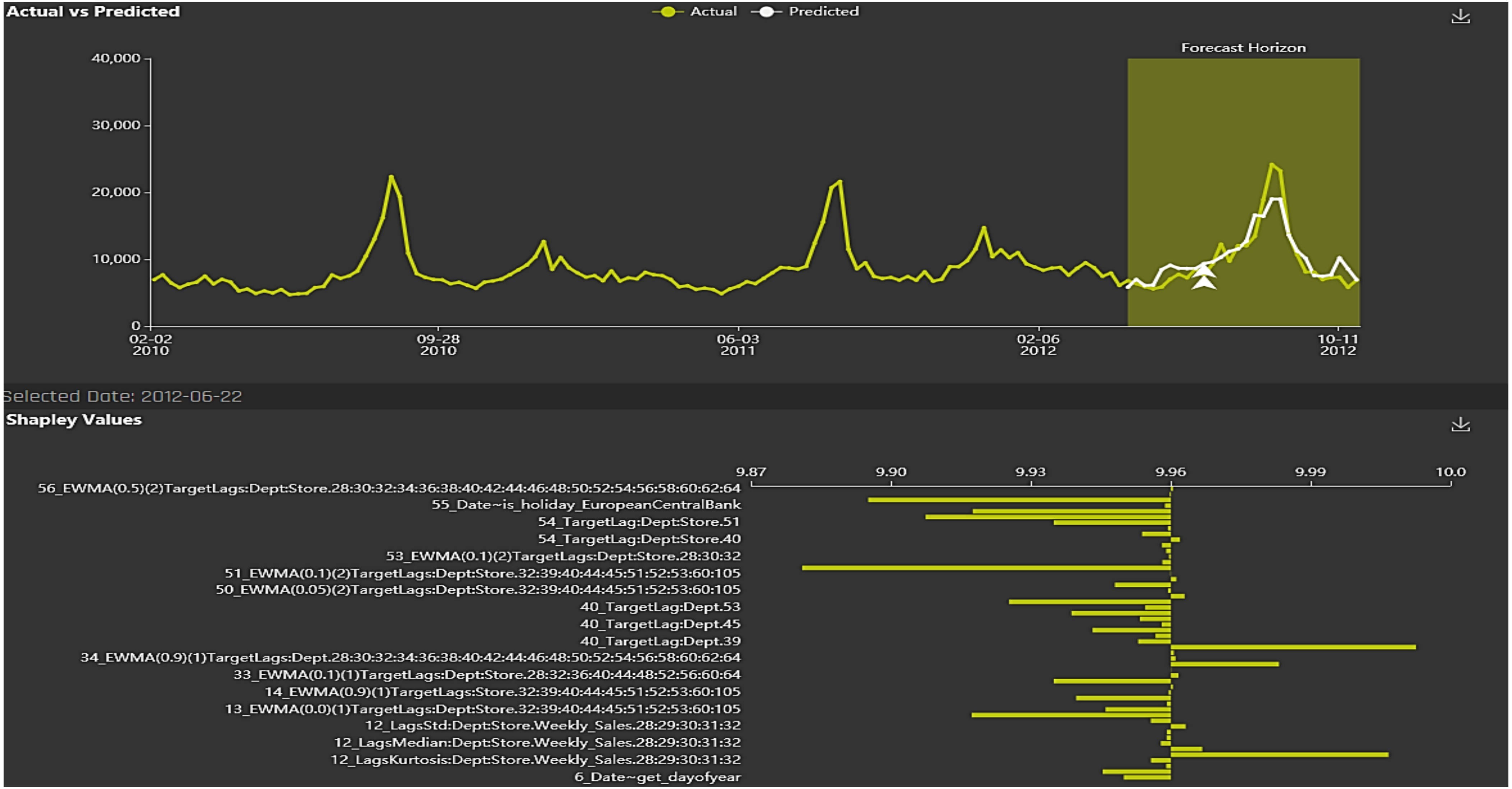
Tuned XGBoost

X% accuracy

Z% accuracy

Feature	importances
lag1(y,x2,x4)	1
lag1(y,x2)	0.5
lag1(y,x4)	0.2
lag1(x1,x2)	0.15 / 0.001
lag1(x3,x4)	0.05

MLI for Time Series



Bring Your own recipe!

- Bring in your domain knowledge and achieve even better results.
- Add additional transformers from the open source git repo : https://github.com/h2oai/driverlessai-recipes
- You can contribute too!
- They follow sklearn type of api
- Add **models**, **scorers** or **transformers**

The Prophet model

y(t)= g(t) + s(t) + h(t)

g(t) Piecewise linear or logistic regressor to calculate **trend**

s(t) models **periodic** changes (e.g. weekly/yearly seasonality)

h(t) **holiday** component

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right)$$

P is the period (365.25 for yearly data and 7 for weekly data)

Parameters [a_1, b_1, ..., a_N, b_N] need to be estimated for a given N to model seasonality.