Is prediction STILL difficult?

Marta Markiewicz

Objectivity
m.markiewicz.pl@gmail.com

September 25, 2019

Abstract

Who wouldn't like to know what future holds? Humans were intrigued with future mysteries since the beginning of time, and so are we at Objectivity. Forecasting revenue is of especially high importance for us, so over last few years we've tried several approaches. First model was productionized in 2016 and it evolved nicely over the years, with various outcomes, coming closer and closer to the actual values. Currently achieved MAPE is around 3%. I would like to share the pros and cons of different methods we've tried (classical time series forecasting, Monte Carlo, ML), to facilitate this journey for others.

I. Introduction

Forecasting has always been a great challenge. Words by Niels Bohr 'Predictions is difficult, especially if it's about the future' capture the essence of the long lasting challenge - the models may fit nicely for the past, they may be even quite good but we literally never know what future will bring us.

For years people have tried to get better and better with prediction. Long known approaches include time series forecasting (Exponential Smoothing, SARIMA) or classical linear approaches (Linear Regression). The era of ML brought to light models like Random Forest or XGBoost. Deep learning also has its part, take for example LSTMs or newly used (M4) ES-RNN. This evolution is just beautiful!

The richness of the models comes from the fact that no model is universal enought to be perfect for all the cases. Doing sales forecasting for quite some time, my observation was that SARIMA with endogenous variables worked best... usually.

The briefing is constructed as follows: it begins with the story of how the classical approaches failed and what was proposed in their place initially. The model evolution description follows, ending with the currently deployed version of the model. It ends with a few pieces of advice.

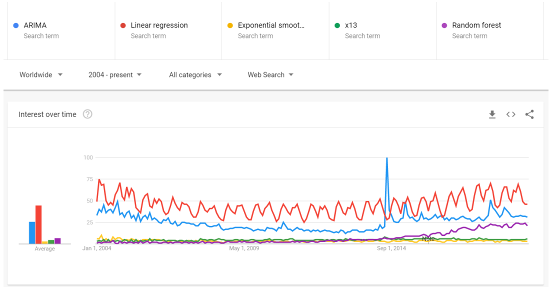


Figure 1: *Classical time series forecasting according to google trend.*

II. Time series approach

At first, we started with classical time series approach. The models are better than wandering in the dark, but they are not immune to sudden changes in the pattern known from history, at least not when speaking about univariate methods. In **??** there is an example of what happens when a suddent behaviour change appears. When being positive, it's a nice surprise but what if such a change strikes us negatively?

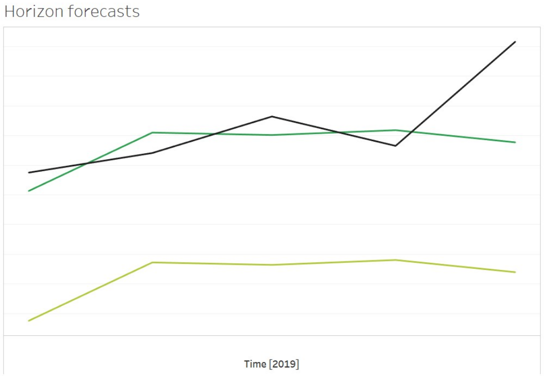


Figure 2: *Classical time series forecasting approach: simple case of failure.*

III. Monte Carlo v1

Seeing the fiasco of classical approach, the way of thinking had to change. Usually there is some hidden knowledge about the future SOMEWHERE. You may not know exactly when you will take holidays, but given you have kids you are pretty sure it will be during their holidays. You don't know your exact expenses for the coming month, but you know car insurance is coming. Sales levels are not known but it's a fact that next week there will be a promotion. In the case of Objectivity there is a sales pipeline for our projects - we don't know what will be sold but we know how many opportunities are in our pipeline and the salesperson has certain 'feeling' about each opportunity success. Why not leverage the knowledge from salesmen heads?

Let denote ith opportunity by $O^{(i)}$, and let's think about it as of the set of parameters:

$$O^{(i)} = \{p^{(i)}, r^{(i)}, \mathbf{e}^{(i)}\} \tag{1}$$

where $p^{(i)}$ denotes probability of an opportunity being a success, $r^{(i)}$ stands for headline day rate and $\mathbf{e}^{(i)} \in \mathbb{R}^H, H \in \mathbb{N}, \mathbf{e}^{(i)} = (e_1^{(i)}, e_2^{(i)}, ..., e_H^{(i)})$ is a vector of an expected effort (team size) for first, second, ..., Hth month. We assumed the horizon to be $H = 6$.

We may also think of $O^{(i)}$ value as of random variable coming from two-points distribution $O^{(i)} \sim \mathcal{D}(p^{(i)}, \mathbf{a}r^{(i)}\mathbf{e}^{(i)})$, $P(O^{(i,h)} = a_h r^{(i)} e_h^{(i)}) = p^{(i)}$, where $\mathbf{a} \in \mathbb{R}^H$ stands for predicted available days for H (6) coming months. Available days are understood as the number of days that a front-office employee dedicates to client work within month (holidays, sickness, ... excluded). Its forecast could be a separate topic, for our purpose let's just assume it's a given number.

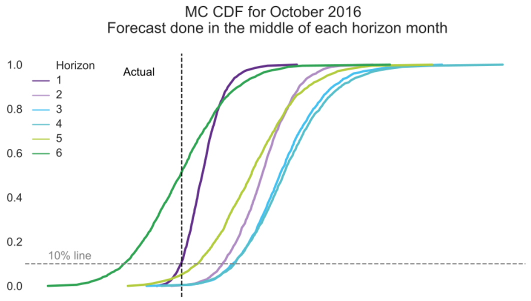


Figure 3: *Monte Carlo simulation example.*

Simultaneously, company revenue for hth month may be defined as $Revenue^{(h)} = \sum_{i=1}^{N} O^{(i,h)}$, where N stand for number of opportunities that are in the portfolio.

Revenue expected value for hth month, becomes then, assuming independence between opportunities:

$$E(Revenue^{(h)}) = \sum_{i=1}^{N} p^{(i)} a_h^{(i)} e_h^{(i)} r^{(i)} \tag{2}$$

The above mentioned definition ignores the existence of dependent opportunities. However, it often happens that a success of one opportunity depends on the other. So the notion of dependency was added through so called *group*. The expected revenue gets more complicated as some opportunities (variables) become conditional on other opportunities.

As model got complicated, we drifted from formulas to Monte Carlo simulations, as they pose no limitation to the flexibility of the model. Also, it empowered us with nice distribution of the sum of not necessary independent random variables.

The exemplary revenue distribution, achieved with 5000 repetitions on 354 opportunities can be seen in figure (see **??**).

Experimentally chosen threshold of 10% was used as forecast. Model was tested in 2015 and moved to production in 2016. Its MAPE for first forecasted month was around 5% - definitely not perfect, but closer to the truth than classical approaches. Please note that theoretically we should have used median and confidence intervals, however tests showed people tend to be overly optimistic... which was actually expected. What's more, the confidence intervals achieved with data input from users were of enormous value for the business as a whole.

As for UI, initially users introduced their input through quite sophisticated Google spreadsheet. Gathered data were processed with JavaScript and the results of MC simulation were shown in the same spreadsheet, just another tab.

IV. Monte Carlo v2

Yet the reality is more complicated. Various opportunities require various skills, they may have several variants with various probabilities, ... And so the model grew. One opportunity O with its probability p was accompanied with opportunity variants, so that rate and effort were no longer constants but they become random variables as well.

Let's denote multipoint distribution by $\mathcal{M}(\mathbf{p}, \mathbf{v})$. $X \sim \mathcal{M}(\mathbf{p}, \mathbf{v})$ satisfies the following conditions:

$$\begin{aligned}
&\mathbf{p} = (p_1, p_2, ..., p_N), \mathbf{p} \in [0,1]^N, N \in \mathbb{N} \\
&\mathbf{v} = (v_1, v_2, ..., v_N), \mathbf{v} \in \mathbb{R}^N \\
&\sum_{j=1}^{N} p_j = 1 \\
&P(X = v_j) = p_j, j \in 1, \ldots, N \\
&E(X) = \sum_{j=1}^{N} p_j v_j
\end{aligned} \tag{3}$$

As a consequence of the above defined notation, an opportunity O becomes:

$$\begin{aligned}
&O \sim \mathcal{D}(p, \mathbf{a}(S) R \mathbf{E}(S)) \\
&\mathbf{a} \in \mathbb{R}^{H+max(S)} \\
&N = |\mathbf{p}^{(rate)}| = |\mathbf{p}^{(effort)}| = |\mathbf{p}^{(start)}| \\
&R \sim \mathcal{M}(\mathbf{p}^{(rate)}, \mathbf{r}), \mathbf{r} \in \mathbb{R}_+^N \\
&\mathbf{E} \sim \mathcal{M}(\mathbf{p}^{(effort)}, \mathbf{E}), \mathbf{E} \in \mathbb{R}^{N \times H} \\
&S \sim \mathcal{M}(\mathbf{p}^{(start)}, \mathbf{s}), \mathbf{s} \in \mathbb{N}_+^N
\end{aligned} \tag{4}$$

and the expected value of it:

$$E(O^{(h)}) = p \sum_{j,k,l} p_j^{(rate)} p_k^{(effort)} a_{h+s_l} r_j E_{k,h+s_l} \tag{5}$$

The reality was, that forecasts worsened! Model was too complicated to explain to everyone, and filling the data became too much of a nightmare. So even if beautiful in theory, we've admitted defeat and took step back.

V. MCML

Monte Carlo in its concept has proven to be extremely useful. The problem was that its initial implementation had the ultimate trust in people while quitting time series theory altogether, so we went from one end of the spectrum to the other. In fourth approach, we decided to merge two ideas.

We considered two possible ways for the improvement: through putting ML on top of MC as well as through probabilities correction. As of today we've initially deployed the first idea, however we still believe in the potential of the second and hopefully we will try it in the future.

i. Experiment design

Let's start with understanding the bigger picture of our experiment design (**??**). The idea was to search the space of data preparation variants $\mathcal{P}$ as well as models space $\mathcal{M}$ to find best possible configuration for our setup (see **??**). It isn't using any trick to find the optimal scenario, like Auto ML or Tpot, as business goal was to produce best possible result for only

ONE time series, so it wasn't computationally challenging. It's key points lie elsewhere, namely it focuses strongly on business success metric definition, as well as smart validation method (**??**).

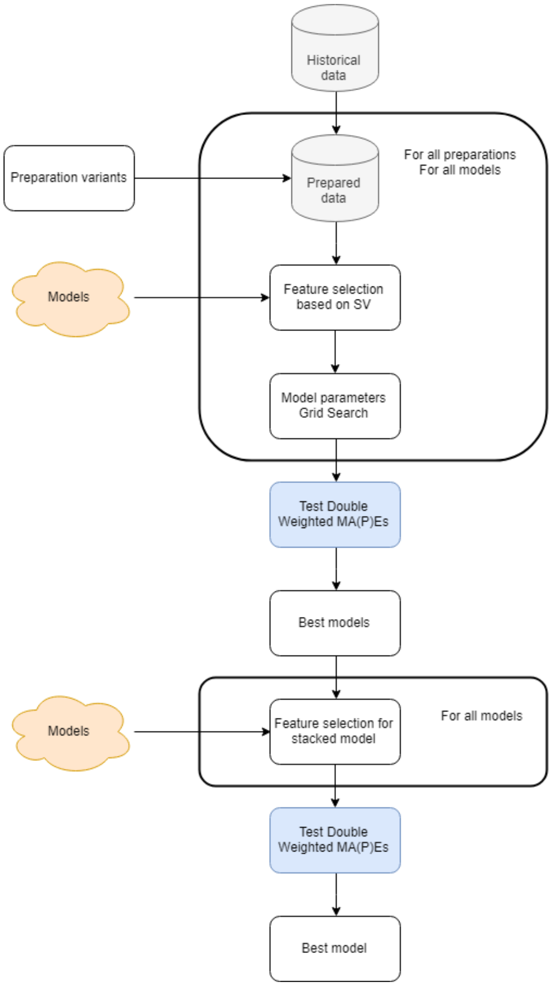

Historical data
Preparation variants
Prepared data
For all preparations
For all models
Models
Feature selection based on SV
Model parameters Grid Search
Test Double Weighted MA(P)Es
Best models
Models
Feature selection for stacked model
For all models
Test Double Weighted MA(P)Es
Best model

Figure 4: *Experiment design for MCML.*

Validation procedure was supposed to mimic the model being on production. It means that:

- each day new data are introduced to the CRM system,
- every day there is a 6 months ahead forecast produced,
- actual, billed revenue values are known with a lag of about 2 weeks in relation to month's end.

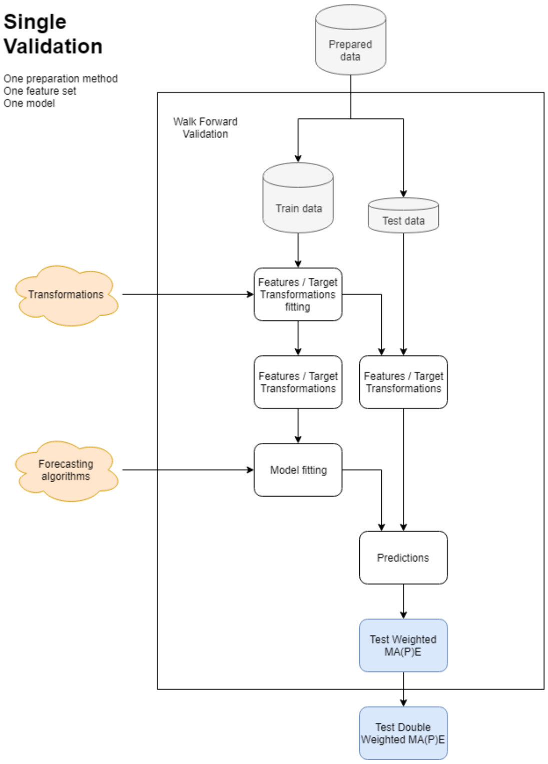


Figure 5: *Single validation MCML.*

Those assumptions would differ for each business case, but it's crucial to reflect them as good as possible in order not to cheat yourself while testing various models. The visualization of train/test split for an exemplary splitting day would look as in **??**.

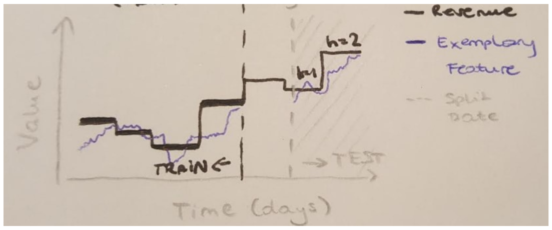


Figure 6: *Walk forward validation with extending window. Please note that that one month is skipped as on production previous month revenue wouldn't be known yet. Also, an exemplary split includes forecast for only two months ahead, as there is no historical data to compare results to for $h \geq 3$.*

ii. Success metric

How to measure the goodness of examined models? MSE, RMSE, MAE, MAPE, ...? Our approach started with understanding business goal priorities:

- accuracy should increase with the decrease of h
- accuracy should increase with every month that model runs on production,
- forecast can not be too optimistic, as it can have disastrous consequences,
- model should be as simple and explainable as possible (as it may happen that we will need to explain it in front of bank guys),
- lower prediction intervals should have nice coverage (Financial Director needs to be prepared for the worse).

Let's denote by $error_i = error(y_i - \hat{y}_i, i, h, day, m, p)$ the ith forecast error for prediction done on p variables, for hth month, on the date day, when m stands from number of months from prediction start till now. Also, let's assume linear weighting:

$$w_{lin}(current, max) = \frac{max - current + 1}{max} \tag{6}$$

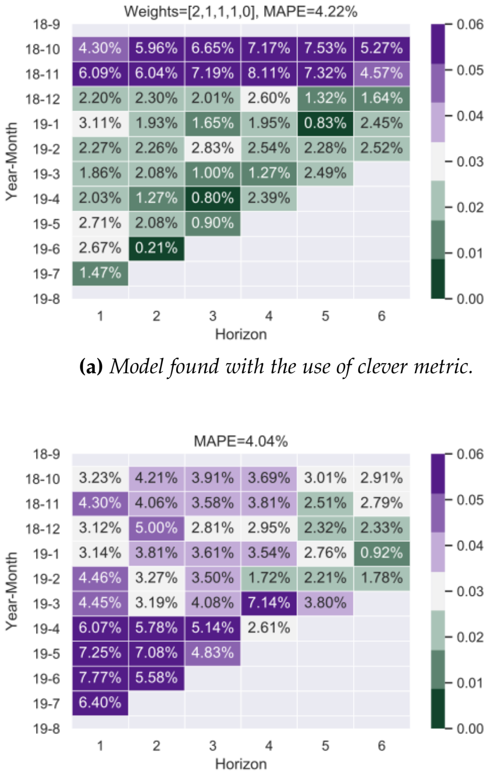
However, it can be any other weight you find suited for your needs.

The above presented priorities were translated into the following set of equations:

$$\begin{aligned}
m_{i,1} &= |error_i| w_{lin}(h, H) \\
m_{i,2} &= |error_i| w_{lin}(m, max(\mathbf{m})) \\
m_{i,3} &= \frac{p}{|feature : features \in \mathcal{F}|} \\
m_{i,4} &= |coverage - assumed\_coverage| \\
m_{i,5} &= \frac{|error_i| \mathbb{1}_{error_i < threshold}}{max_{i \in I}(|error_i| \mathbb{1}_{error_i < threshold})}
\end{aligned} \tag{7}$$

If $error \in [0,1]$, then $m_1, ..., m_5 \in [0,1]$ which makes it easy to define success metric:

$$m = \frac{1}{|I|} \sum_i \sum_j w_j m_{i,j} \tag{8}$$



(a) *Model found with the use of clever metric.*

(b) *Model found with simple MAPE.*

Figure 7: *Metrics comparison - sophisticated weightes (a) vs MAPE (b). Note that even the MAPE is smaller for (b), the distribution of it differs more favourably for (a).*

where $\sum_{j=1}^{5} w_j = 1$. This way $m \in [0,1]$, which gives very nice interpretation. Another possibility uses product instead of sum:

$$m = \frac{1}{|I|} \sum_i \prod_j m_{i,j}^{w_j}, \tag{9}$$

where $w_j \in \mathbb{R}_+$.

iii. Results

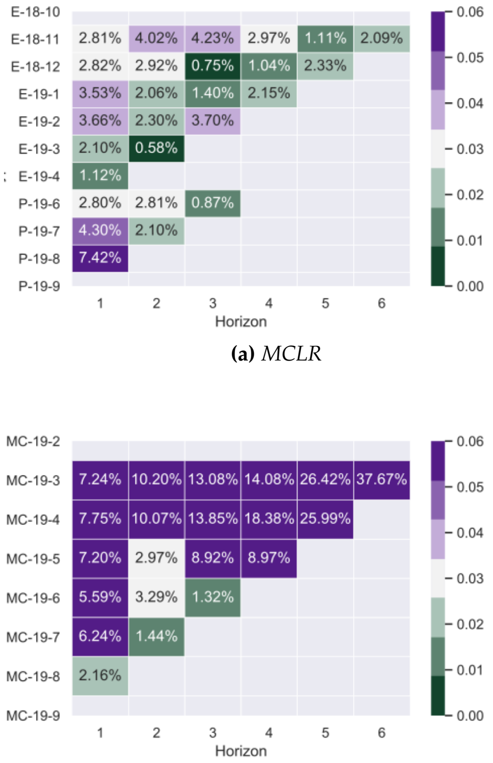
Using the above described Experiment design and Success metric, the space of proposed features combination, models and stacked models has been searched through. Checked features were divided into groups:

- seasonal (year, quarter, month, day of week, etc.):

  - seasonality of predicted month,
  - seasonality of input time,
- input values:
  - effort, rate and probability buckets,
- Monte Carlo related:
  - spreads,
  - percentiles,
  - percentiles from the past,
- Revenue from the past, etc.

Depending on the time aggregation, various models have been checked: SARIMAX, Exponential Smoothing, LSTMs, Prophet, Random Forest, XGBoost, SVM, Linear Regression, Tpot. Interestingly enough, Linear Regression was tried out only for the purpose of being a benchmark against more innovative algorithms. However, numbers were merciless - it was second to none. While for LR it was easy to achieve MAPE around 3%, for others it was a struggle. Random Forest achieved 6.7%,SARIMAX 5.2%, even Tpot only approached 4%.

Figure **??** presents results for MCLR (MC with Linear Regression), both results of experiments and current performance on semi-production and production.



(a) *MCLR*

(b) *MC Model*

Figure 8: *Currently and historically deployed model performance.*

VI. Conclusion

To conclude, I have few pieces of advice for you:

- Don't be afraid to start small. As long as it's not harmful, it's better to have simple solution but on production, that end up with pure research.
- Leverage expert knowledge. Domain knowledge in people heads runs the world for years, use it to your advantage. Cooperate with knowledge experts rather than racing with them.
- Challenge any additional complexity in manual data collection. Gain may not be worth it.
- Choose success metric wisely. Of-the-self classics may not be suited for your needs.
- Check simple models along with state-of-the-art. Worse case scenario, you will have benchmark values. Best - they may surprise you.

Enjoy forecasting!