RdR score: Proposing a new experimental technic for evaluating time series forecasting models

In this text, I will propose you an experimental technic to evaluate the performance of time series forecasting models. This new technique will give several benefits such as being able to:

- Compare models together and select the best one
- Facilitate explanation to manager or business team
- Help decide whether the forecasting model should be use or not
- How much a forecasting model is good, alone or compared to other models
- Use the ***shape similarity*** of the forecast as an important evaluation attribute
- Use ***randomness*** as an important evaluation criterion; is the forecasting model better than a naïve random decision? How much better?

The proposed **RdR** metric use:

- **R**: Naïve Random Walk
- **d**: Dynamic Time Warping
- **R**: Root Mean Squared Error

Warning: This is very experimental and not issued from any research paper. I named this technic RdR only to give this experimentation a name and to facilitate the understanding of what the metric actually do. Use it at your own risk!

The code of this experimental RdR score technic is available on my github.

To experiment this metric, we will work on three different datasets and four models to resolve multistep forecasting problems:

- SARIMA (Box-Jenkins method)
- Holt-Winters (Triple Exponential Smoothing)
- LightGBM (Gradient Boosting) (as a Multivariate Multi-Target Regressor)
- Seq2Seq (Deep Learning) (as a Multivariate Multi-Target Regressor)

I will assume that you know those models. If not, those models are a mix of popular old school econometric forecasting, machine learning and deep learning. There is plenty information on them if you search on Google.

Currently, the most famous metrics for evaluating time series forecasting models are MAE, RMSE and AIC.

I will assume that you know those metrics. If not:

- MAE vs RMSE: https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d
- AIC: https://en.wikipedia.org/wiki/Akaike_information_criterion, https://stats.stackexchange.com/questions/24116/one-sentence-explanation-of-the-aic-for-non-technical-types, https://towardsdatascience.com/the-akaike-information-criterion-c20c8fd832f2

To briefly summarize, both MAE and RMSE measures the magnitude of errors in a set of predictions. The major difference between MAE and RMSE is the impact of the large errors. For example, if some prediction data points are large outliers errors when compared to the ground truth, those large errors will be diluted in the mean of MAE while RMSE score will be higher because of the square operation.

AIC measure the loss of information and penalize the complexity of a model. It is the negative log likelihood penalized for a number of parameters. The main idea behind AIC is that model with less number of parameters is better. AIC lets you test how well your model fits the dataset without overfitting it.

	Value range	Interpretation	Limitations
MAE	0 to ∞	The lower, less are the errors, better is the model	Use to compare model on the same dataset only. Alone, difficult to decide whether your model is good or not.
RMSE	0 to ∞	The lower, less are the penalized errors, better is the model	Use to compare model on the same dataset only. Alone, difficult to decide whether your model is good or not.
AIC	-∞ to ∞	The lower, less are the information loss and model complexity, better is the model	Use to compare model on the same dataset only. No possible interpretation if used alone.

With MAE and RMSE, the perfect score is 0 (The goal is to have the lowest score possible). Both values range from 0 to ∞, and depend on the scale of the target we want to forecast.

MAE is the easiest to interpret. For example if MAE is 450$, we could say that our forecasting model have an average error of 450$ per forecast. Very easy to explain to a manager or a business team. Is it good? Should we use it? Well, it depends on the use case context, the distribution of the errors (skewed or not, outliers or not) and many other things.

RMSE interpretation is less intuitive. If RMSE is 450$, we could say that our forecasting model have an average “penalized” error of 450$ per forecast, which is somewhat not very intuitive.

Some might sell it as the equivalent of the average error but adjusted for stability and large gap in predictions (adjusted MAE)

For AIC, there is no perfect score, the lower is the better. Soo it is not possible to evaluate the AIC score alone, it is only use to compare models together. For example, if used alone, an AIC of -950 does not give any clue about the model performance and impossible to explain to a manager.

A last one, the **Mean Forecast Accuracy:** Calculating the mean forecast accuracy is also an interesting metric. This metric is very intuitive and easy to explain to a manager (Our model have an average of 66% forecasting accuracy, which also means that our models have an average error of 34%). It gives a good approximation of how well a forecasting model performs. For example, take these forecasting results:

Y_TRUE	Y_FORECASTED	ABSOLUTE ERROR	FORECAST ACCURACY CALCULATION	FORECAST ACCURACY
50	48	2	= GREATEST(1 - (2 / 50), 0)	96%
35	60	25	= GREATEST(1 - (25 / 35), 0)	29%
40	40	0	= GREATEST(1 - (0 / 40), 0)	100%
42	39	3	= GREATEST(1 - (3 / 42), 0)	93%
25	250	225	= GREATEST(1 - (225 / 25), 0), 0)	0%
22	27	5	= GREATEST(1 - (5 / 22), 0)	77%
Mean Forecast Accuracy:			**=MEAN(FORECAST_ACCURACY)**	**66%**

Well... Nothing is perfect!

We can see that this metric have a **major flaw**. The mean has one main disadvantage; it is particularly susceptible to the influence of outliers. When the forecast results are really bad (when the error alone is higher than the expected ground truth result), the percentage can goes really low (In this example: (1 – (225/25)) gives ***Minus 800%***) which will have a big negative impact on the global Mean Forecast Accuracy.

- A solution to this problem could be to **clip the minimum percentage value** to 0%, to reduce the impact of isolated / outlier forecast results.
- We could also have use the median instead of the mean, but the result could be more diluted than if we clip to zeros.

For example, if we used median instead of zero clipping, it gives us a Median Forecast Accuracy of 85% (the outlier have been ignored), compared to 66% with Mean Forecast Accuracy with zero clipping.

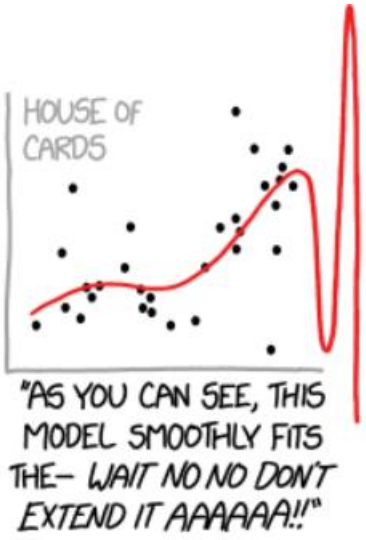
	FORECAST ACCURACY	FORECAST ACCURACY CLIPPED
	96%	96%
	29%	29%
	100%	100%
	93%	93%
	-800%	***0%***
	77%	77%
MedianForecastAccuracy	***85,06%***	***85,06%***
MeanForecastAccuracy	***-67,55%***	***65,78%***

In general, when your error distribution is skewed, you should use Median instead of Mean. In some case, the Mean Forecast Accuracy could also mean totally nothing. If you remember your statistics; the coefficient of variation (CV) represents the ratio of the standard deviation to the mean (Coefficient of Variation = (Standard Deviation / Mean) * 100). A big CV value means big variability, which also means greater level of dispersion around the mean. For example, we could consider anything above a CV of 0.7 as highly variable and not really forecastable. In addition, it can show that your forecasting model prediction's ability is very unstable! (The mean have a poor meaning in this case) (source: https://blog.arkieva.com/do-you-use-coefficient-of-variation-to-determine-forecastability/)

Okay, before we deep dive into the experimental **RdR** technic, I would like to talk about one important mistake I have seen so many times on the internet: ***Please, do not calculate performance metrics on the fitted data / training data!***

In ***traditional statistics***, time series forecasting models were often evaluated on the "fit" ("find the best fit") predictions results of the model, which I think, makes absolutely no sense at all! In machine learning process, are we evaluating / choosing our models on the fit results? ***NEVER (I hope so!)***. Unless you want to know if your model is overfitting or underfitting (you can compare validation and training set results)

I think this way of thinking comes from old fashion curve fitting technics where the goal is to fit a curve as best as possible. When you add the time dimension, the problem is that when we try to **extrapolate** this overfitted curve through time, the results are rarely good:



Source: https://www.explainxkcd.com/wiki/index.php/2048:_Curve-Fitting

Another big problem with that way of thinking is, when you use advanced technics like gradient boosting ensemble or deep learning, the fit will usually be very good or perfect and hide overfitting problems that you will only discover in the ***extrapolation process through time***.

I have read several experiments where a deep learning model is perfectly fit on a time series with the conclusion: ***"Wow, deep learning with time series is so revolutionary and incredible!"*** (And of course, you don't see the extrapolation process through time that should come next and that probably didn't work very well…). Fitting is not a problem anymore (You just have to set a big number of estimators to an ensemble model or a big number of epochs to a neural network model), overfitting is! Time series forecasting models should never be validated differently than standard data science regression problems (The main difference is that you train your model to extrapolate "n steps" into the future instead of interpolating a value in a multivariate values range domain). In plain words, if you are doing forecasting, only calculate your performance scores / evaluate your models on period(s) of time (in the future) that the model never seen before to fit the data (Unless your investigating for underfitting / overfitting problems)

OKAY, NOW WE CAN GO!

The proposed RdR score technic will mainly answer three questions:

How can we take into account the shape similarity of a time series? :

- Answer: **Dynamic Time Warping**

How can we know if we should use our forecasting model or not? :

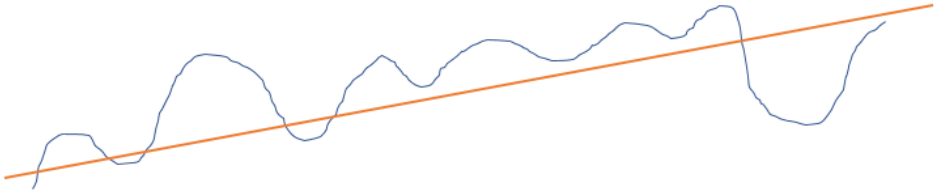
- Answer: Is it better or worst than a **Naïve Random Walk?**

How can we take into account the errors? :

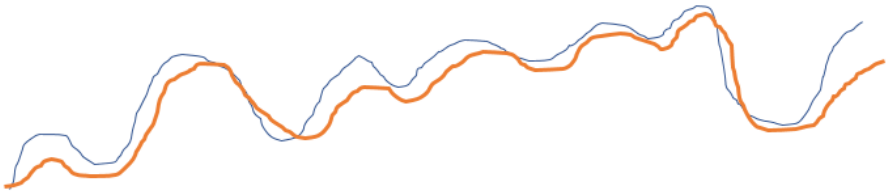
- Answer: **Root Mean Squared Error (RMSE)**

Now, suppose we have those two following models, with exactly the same RMSE:

Model A, RMSE = 350 (On test set / unseen period in the futur)



Model B, RMSE = 350 (On test set / unseen period in the futur)



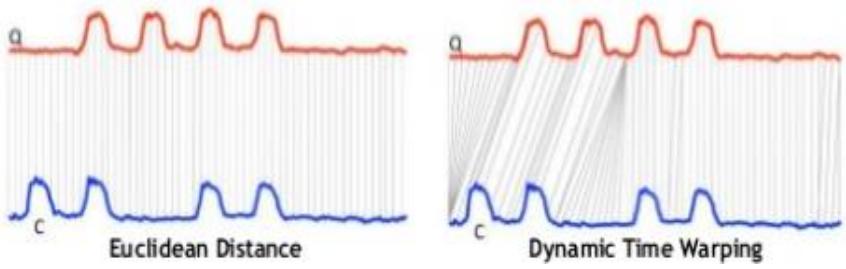
Which model would you choose?

One big flaw about using MAE, RMSE or AIC; those metrics does not take into account an important attribute of the forecast: **THE SHAPE SIMILARITY!**

Why use Dynamic Time Warping (DTW) as a similarity metric? :

- Euclidean distance between time series: Bad choice because there is distortion in the time axis
- DTW: Find the optimal (minimum distance) warping path between two time series, by « synchronizing » / « align » the different signals on the time axis
- The DTW distance is the square root cost of optimal warping path

- The lowest the distance on the warping path, the more similar are the time series.

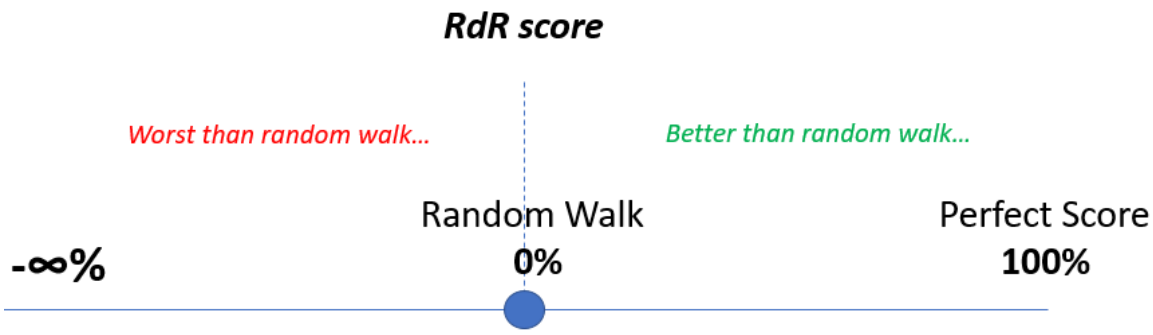


If you want to know more about Dynamic Time Warping (DTW): https://www.slideshare.net/DavideNardone/accelerating-dynamic-time-warping-subsequence-search-with-gpu

When a manager ask you, "should we use our forecasting model to predict the future?" An interesting answer could be: Well, based on the performance evaluation metrics, our model is 65% better than if we randomly make a decision. This is where random walk shows off.

	Value range	Interpretation
RdR	-∞% to 100%	**The greater near 100%, the better.** **Positive score:** Better than a random walk **Negative score:** Your model is worst than a simple random walk model. You should not use it! **If your model is 0**: Your model is equal to random walk performance. **Perfect score**: 100, your model is equal to the ground truth, the time series are equal in all aspects.

We can interpret the RdR score as the percentage of difference (In error and shape similarity) between your model and a simple random walk model, based on RMSE score and DTW score. If the percentage is negative, your model is ***[X]%*** worst than randomness. If the percentage is positive, your model is ***[X]%*** better than randomness. In another words, the ***[X]%*** will change depending on the RMSE errors and the DTW distance around 0, which is the bound of naïve randomness. Why RMSE instead of MAE? I think that penalizing the average error with large errors reflect more the reality and the stability of the model.

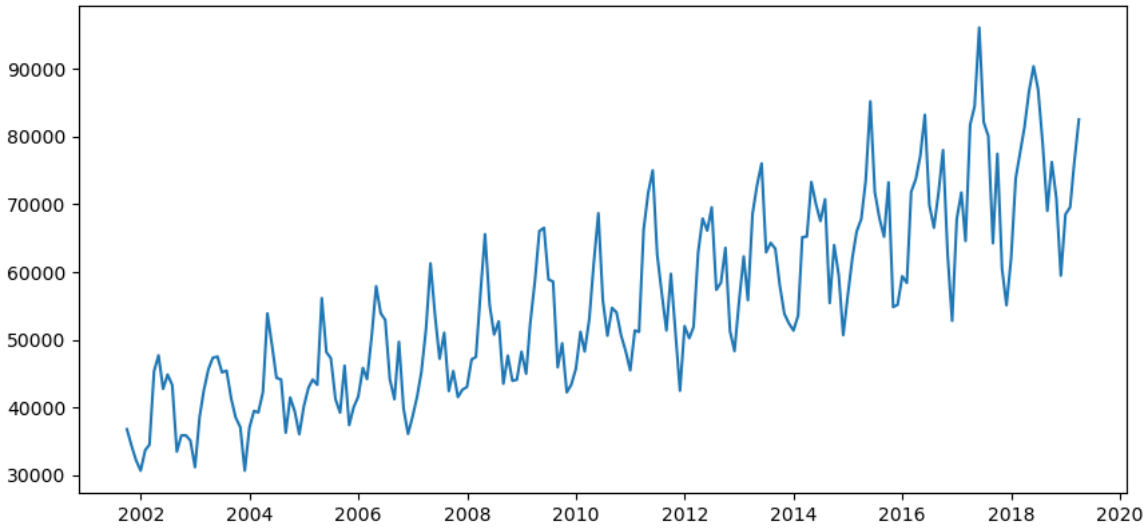


-∞% means that sometimes, you may have a very big negative percentage score like **-98695%**; This is bad news and it means that you should definitely not use the model!

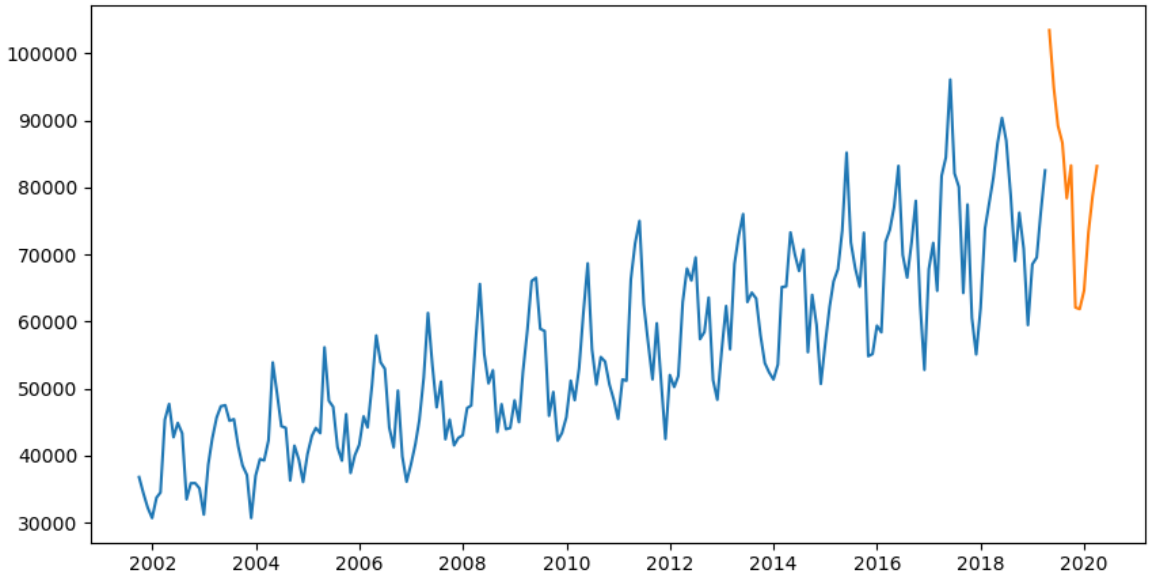
There could also be a strange situation where the perfect score equals the random walk model (A straight line). Well, in his case, you do not really need a model!

Enough talking, let's try it and see what happens!

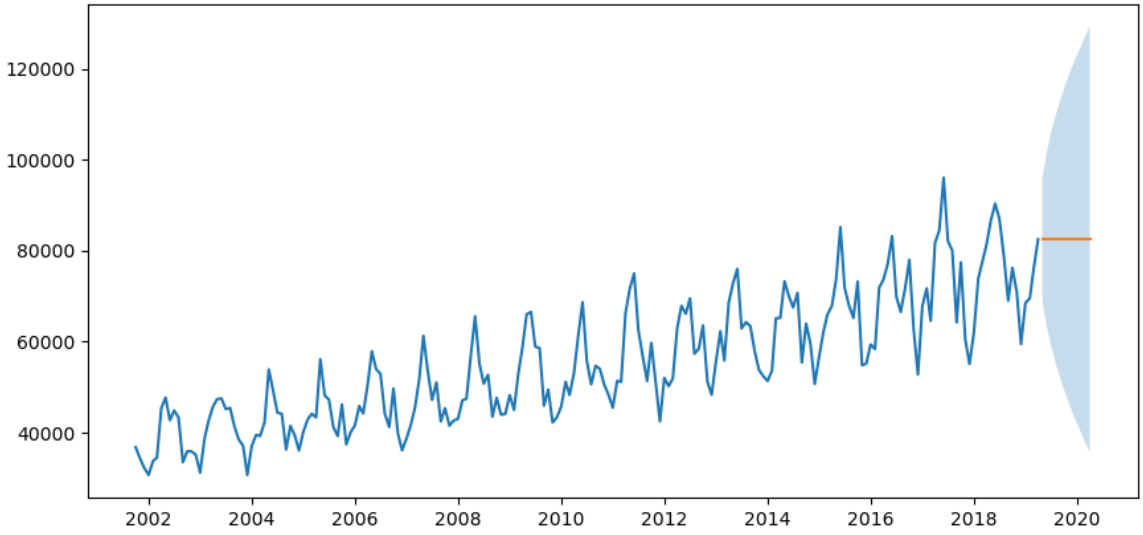
Experiment Dataset#1: Easy (Good autocorrelation and seasonal structure, deterministic system):



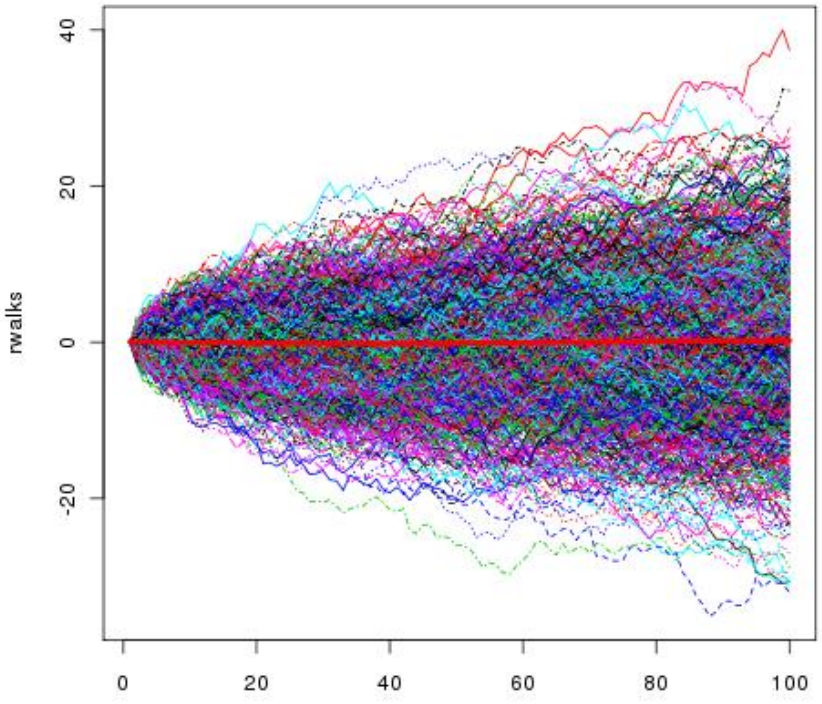
The multistep unseen test data we will use to validate the performance of our models:



The naïve random walk model (RdR Score = 0):

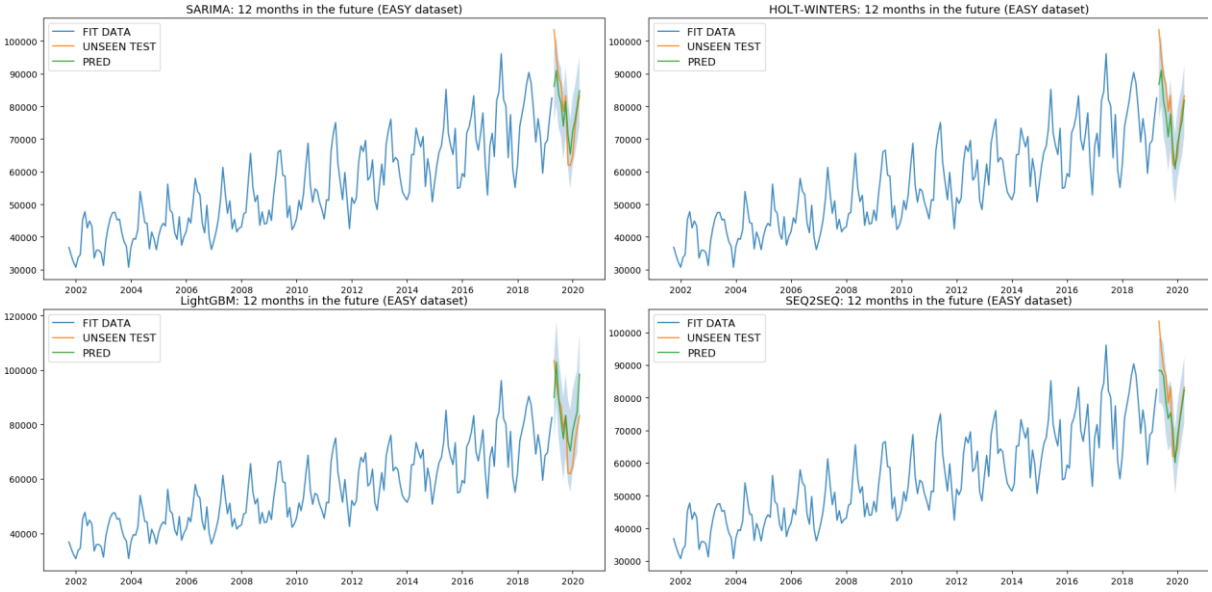


The forecasts from a random walk model (without drift) are equal to the last observation; Scenario where future movements are unpredictable and are equally likely to be up or down (Stochastic). It gives a straight line because if we do infinite random simulations from the last data point, as the chances are equally likely to be up or down, the mean will be equal to the last observation. Like this:

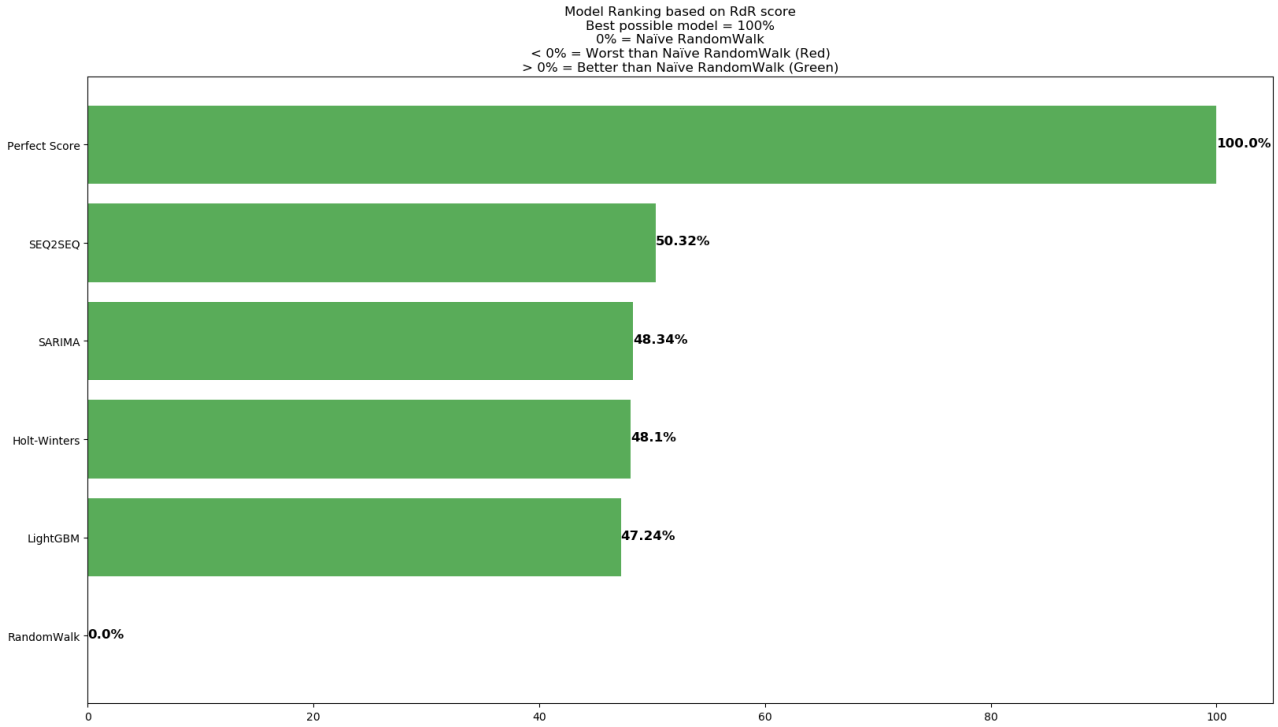


Can we beat this?

Here, the multistep forecasts of 12 periods in the future (unseen test data) of our four models:

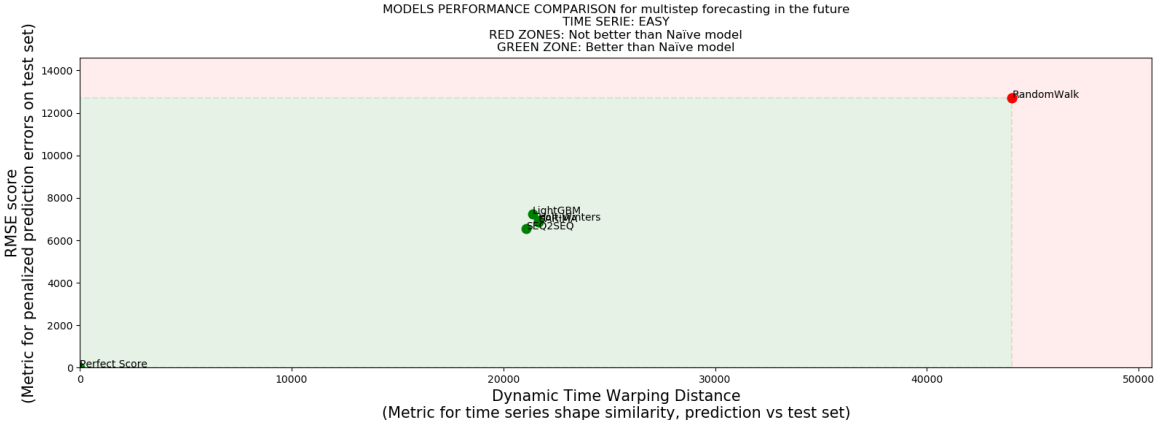


Let's calculate the RdR score of each model:



With this graph, we can see that all the performance are very near. The best model is Seq2Seq.

If we want to have more detailed information on the RdR score, we can plot the RMSE vs DTW like this:



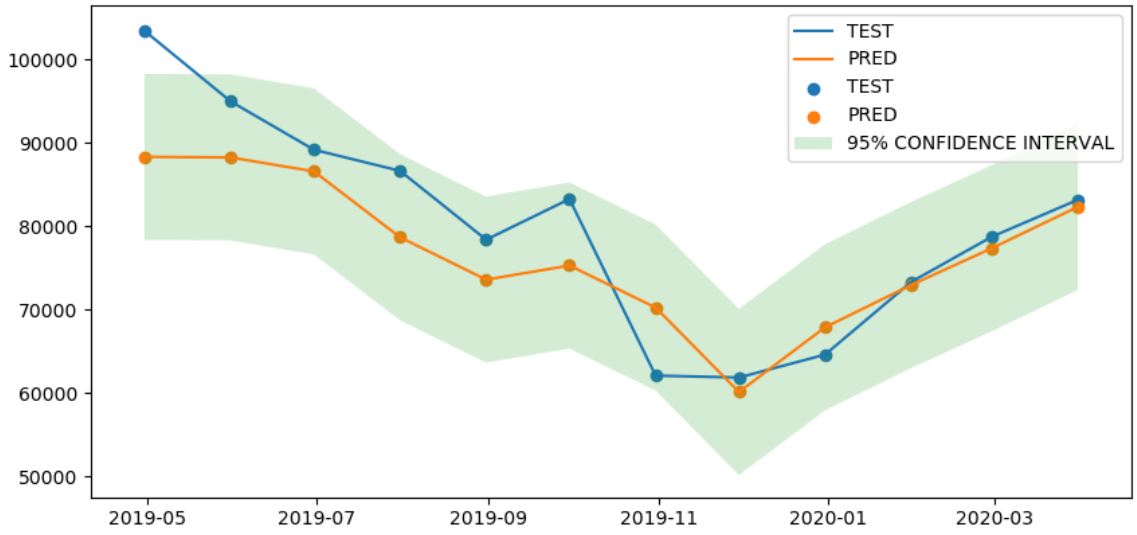
The y axis is for the penalized errors while the x axis is for the shape similarity of the time series (between prediction and unseen test dataset). In this graph, we can see that Seq2Seq was the best model, in both error and shape and that it is halfway between the random walk score and the perfect score (which represent the 50,32% RdR score). We can see that LightGBM have slightly more errors than the econometric models (RMSE) but have a slightly better shape

similarity (DTW distance). All models were a lot better than the Random Walk model performance.

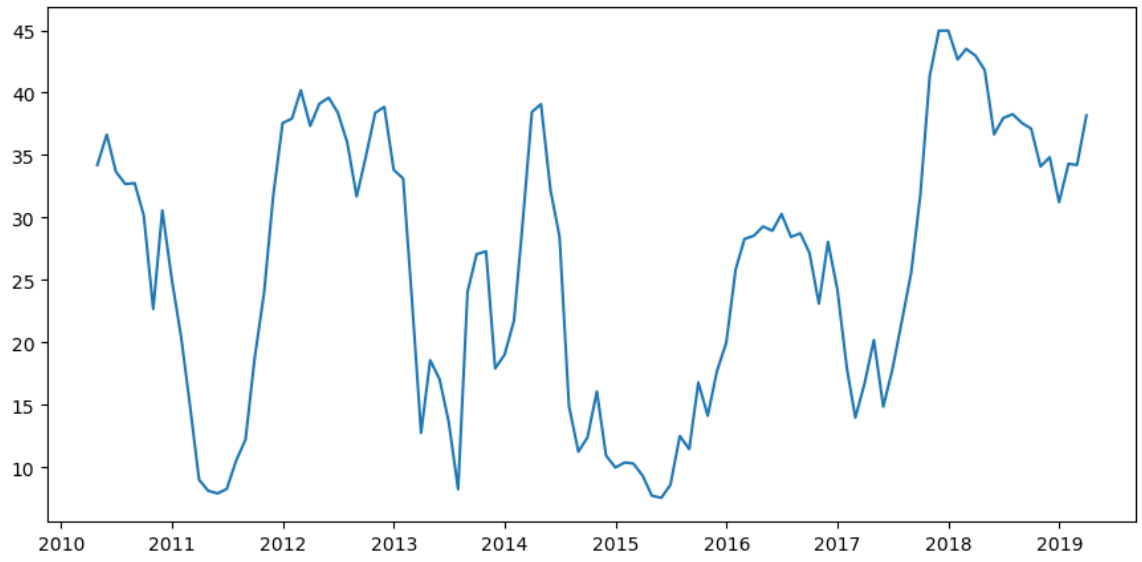
Interpretation of the best model (Seq2Seq):

```
In [504]: rdr_score4.get_rdr_interpretation()
Out[504]: 'GOOD PERFORMANCE:With a stable trend and no major unpredictable changes,
the model is 50.32% better than a naïve random decision. The mean forecast accuracy is
91.61% (around 89.42% and 93.67% per datapoint)'
```

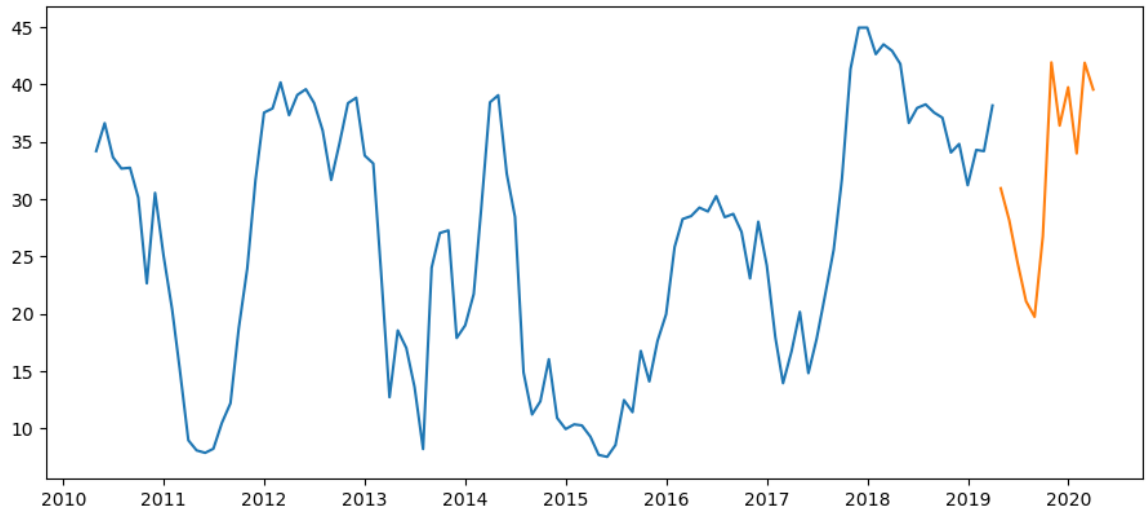
If we zoom the prediction of the best model (Seq2Seq):



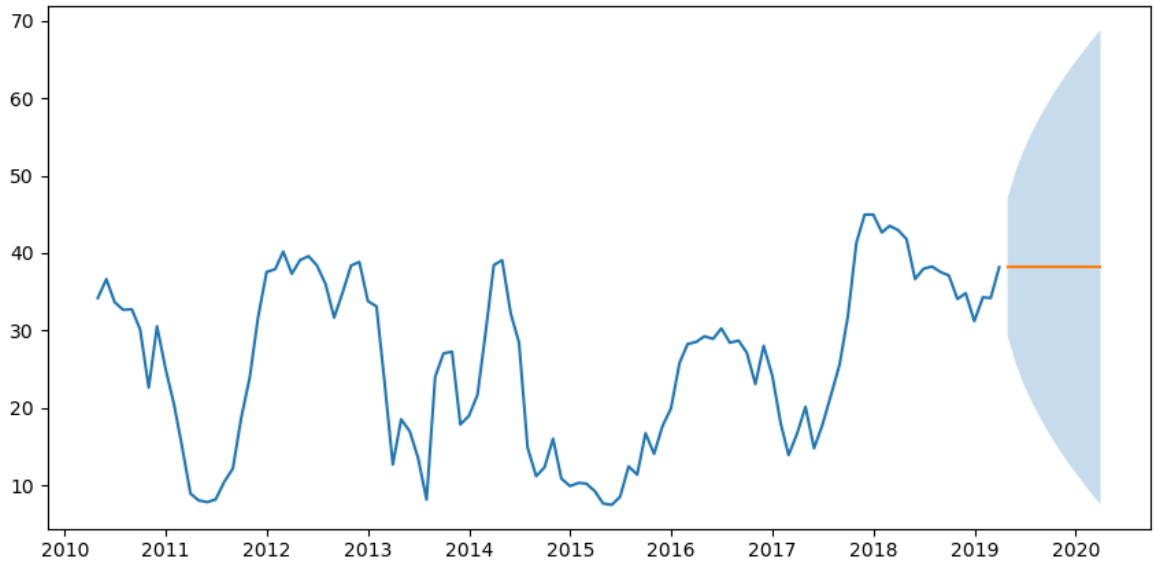
Experiment Dataset#2: Medium (Autocorrelation and seasonal structure, with a lot of noise, hybrid deterministic-stochastic system):



The multistep unseen test data we will use to validate the performance of our models:

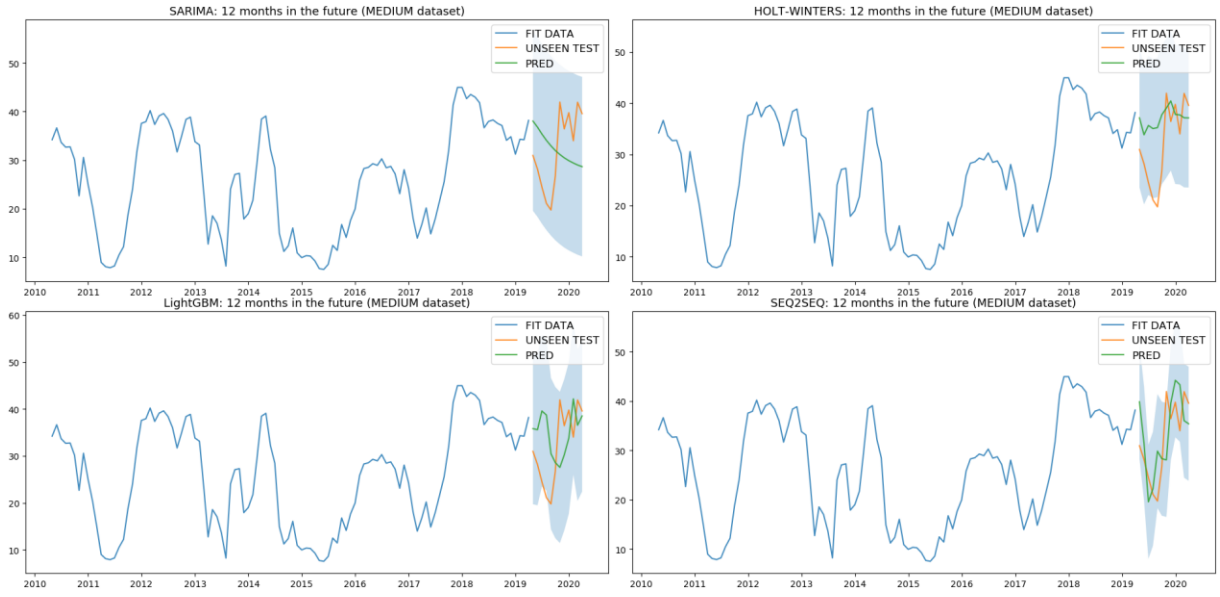


The naïve random walk model (RdR Score = 0):

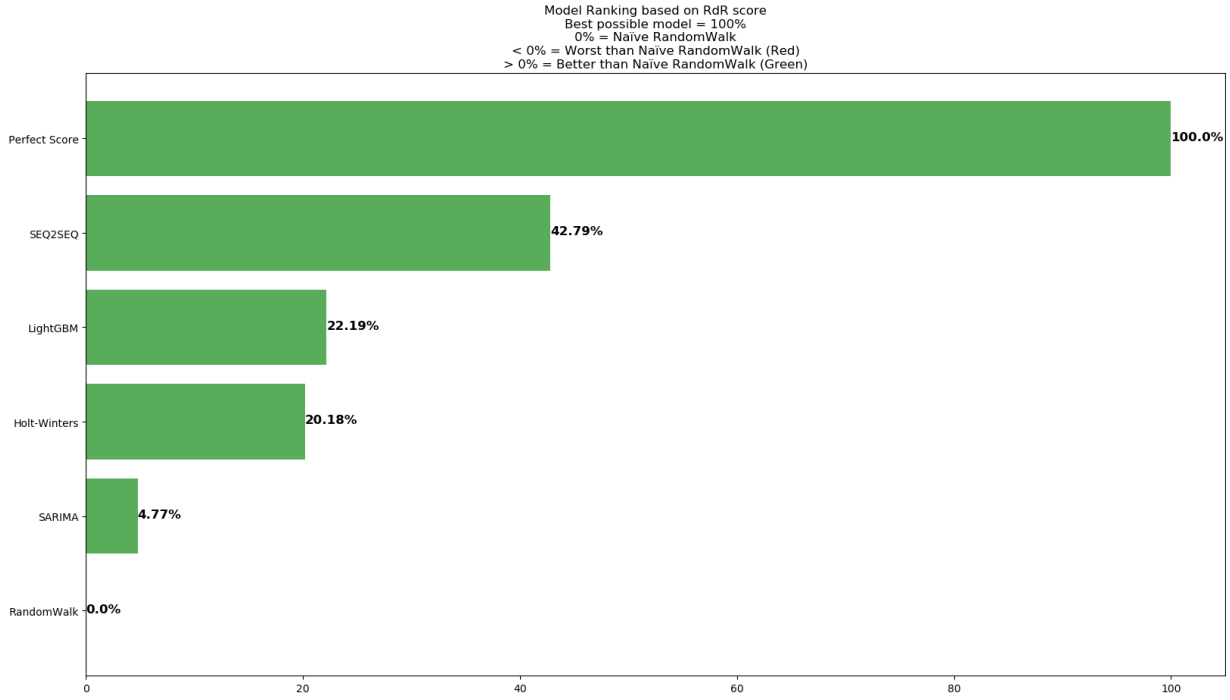


Can we beat this? :

Here, the multistep forecasts of 12 periods in the future (unseen test data) of our four models:

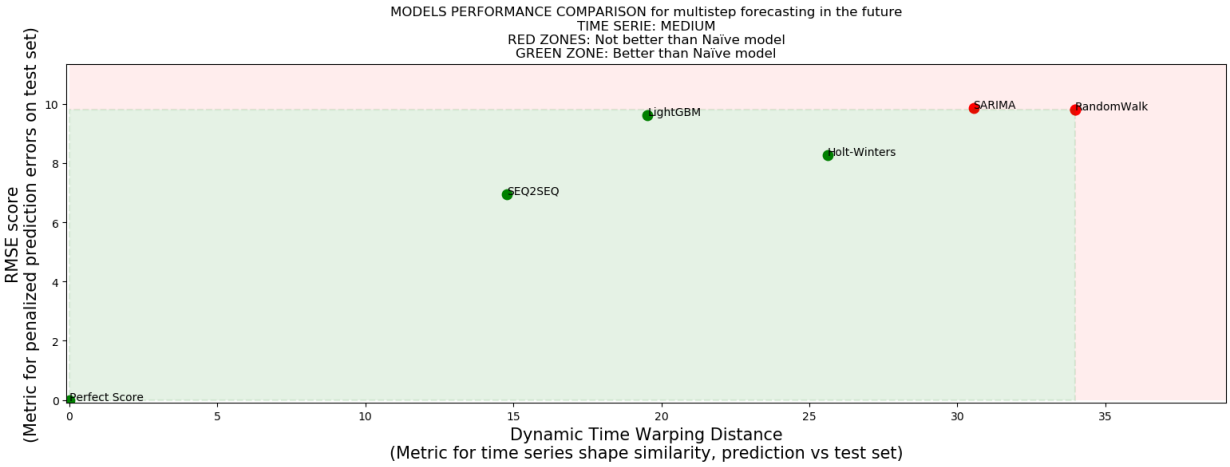


Let's calculate the RdR score of each model:



We can see that the SEQ2SEQ model was the best. SARIMA performed the worst but all models are still better than the random walk. As we expected, we can also see that globally, this round was more difficult than the previous one; the RdR scores are lower.

If we want to have more detailed information on the RdR score, we can plot the RMSE vs DTW like this:

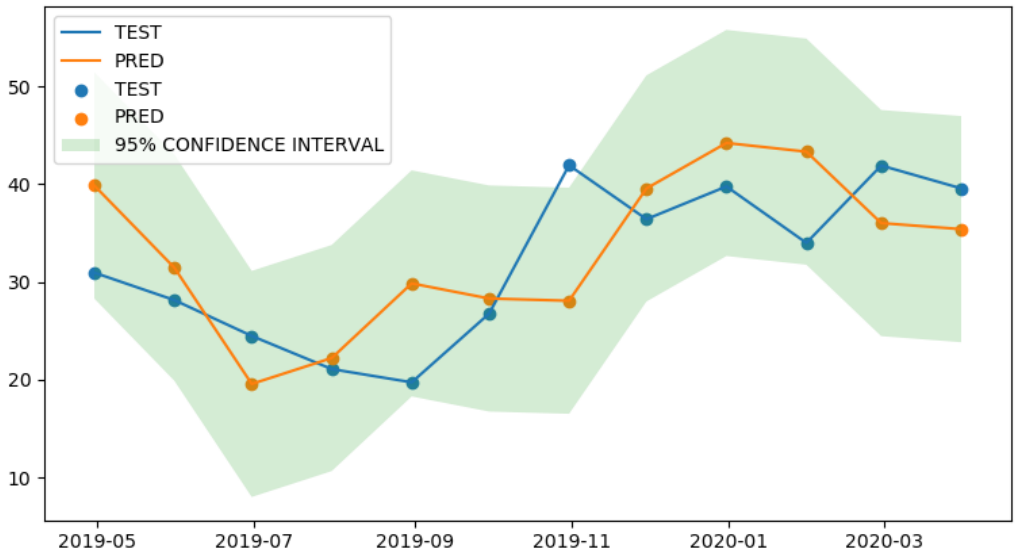


The y axis is for the penalized errors while the x axis is for the shape similarity of the time series. In this graph, we can see that Seq2Seq was the best model, in both error and shape. SARIMA was almost equal RMSE than random walk but the shape of SARIMA time series was better (DTW distance).

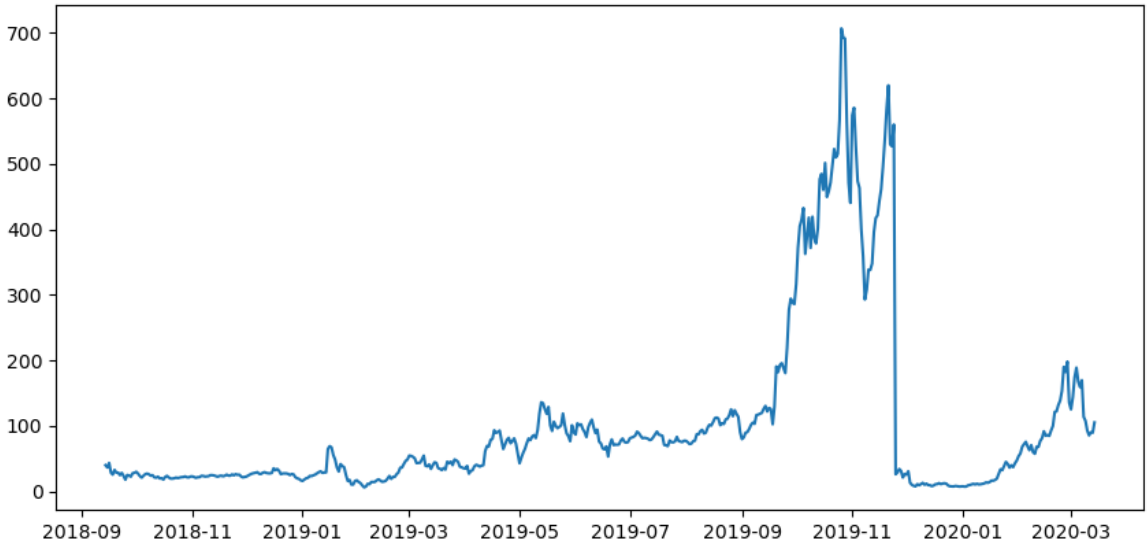
Interpretation of the best model (Seq2Seq):

```
In [469]: rdr_score4.get_rdr_interpretation()
Out[469]: 'GOOD PERFORMANCE:With a stable trend and no major unpredictable changes,
the model is 42.79% better than a naïve random decision. The mean forecast accuracy is
76.86% (around 64.76% and 83.41% per datapoint)'
```

If we zoom the prediction of the best model (Seq2Seq):



Experiment Dataset#3: Hard (No Autocorrelation, no seasonal structure, stochastic system) – Stock Price alike:

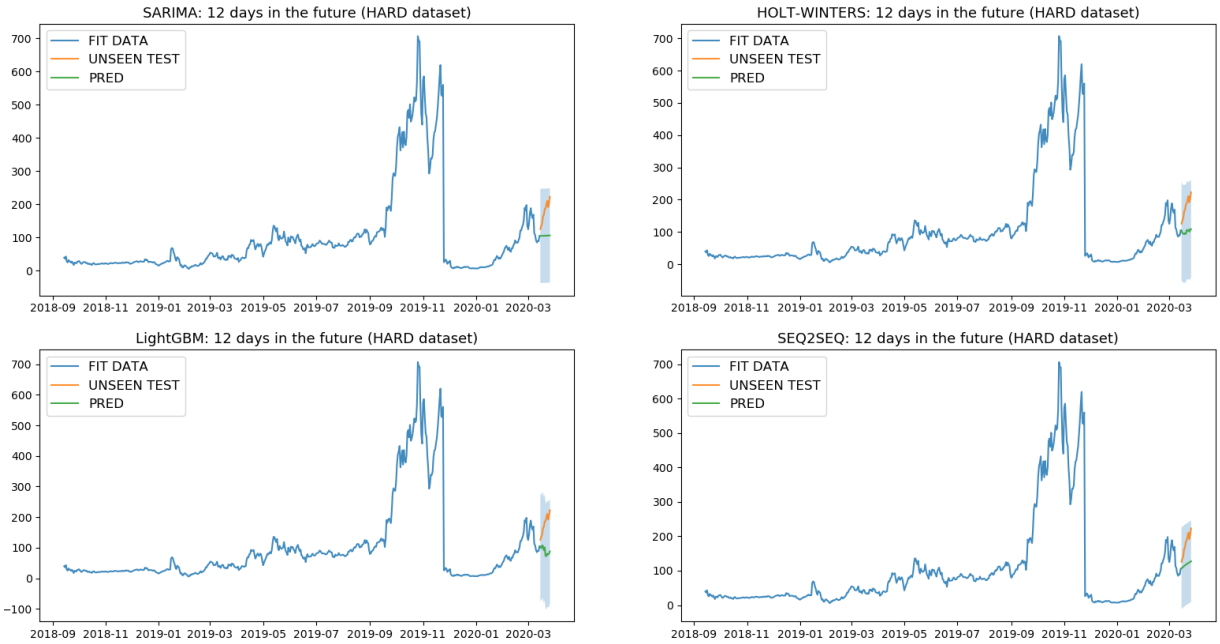


The multistep unseen test data we will use to validate the performance of our models:

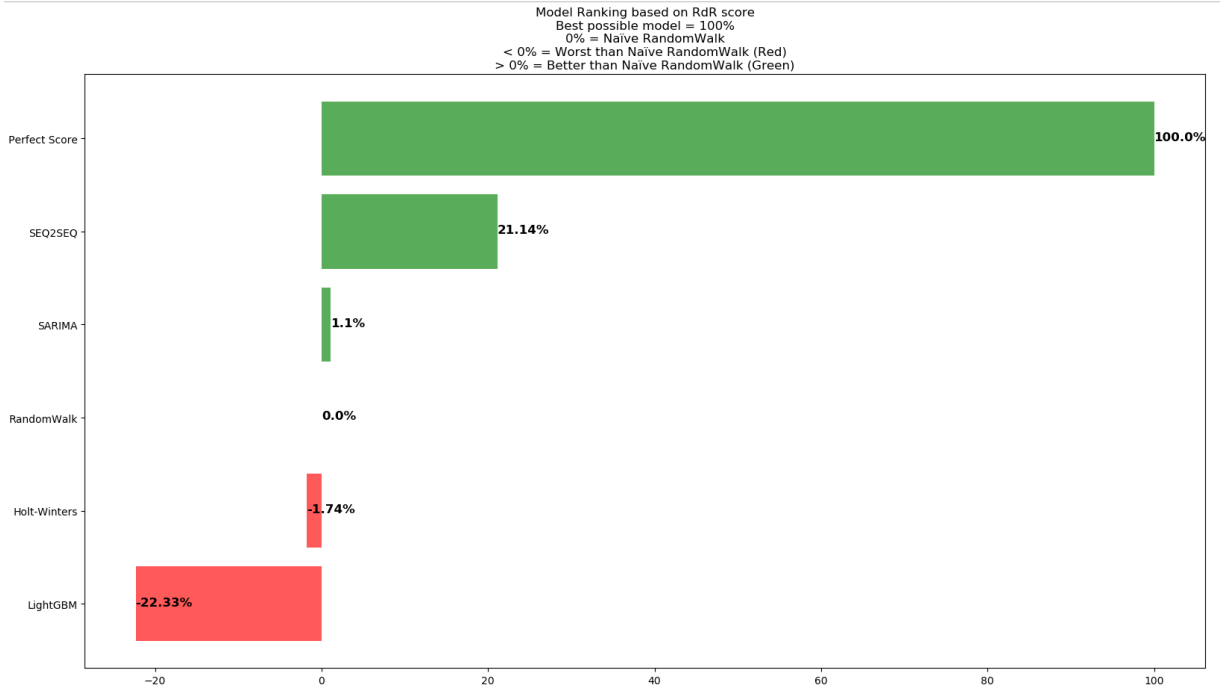
The naïve random walk model (RdR Score = 0):

Can we beat this?

Here, the multistep forecasts of 12 periods in the future (unseen test data) of our four models:

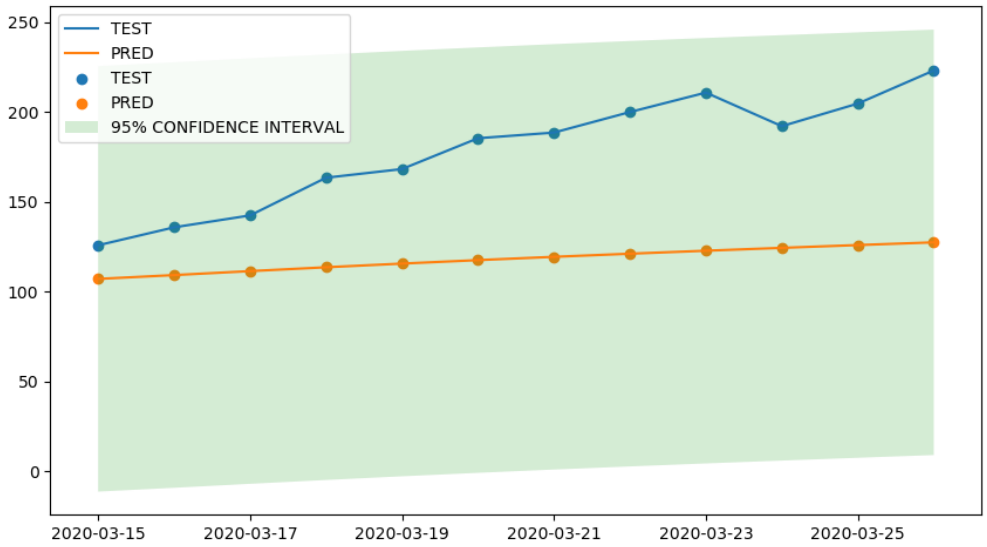


Let's calculate the RdR score of each model:



Just by looking at this bar chart, we should not use both Holt-Winters and LightGBM as they are worst (in error and shape similarity) than a simple naïve random walk model.

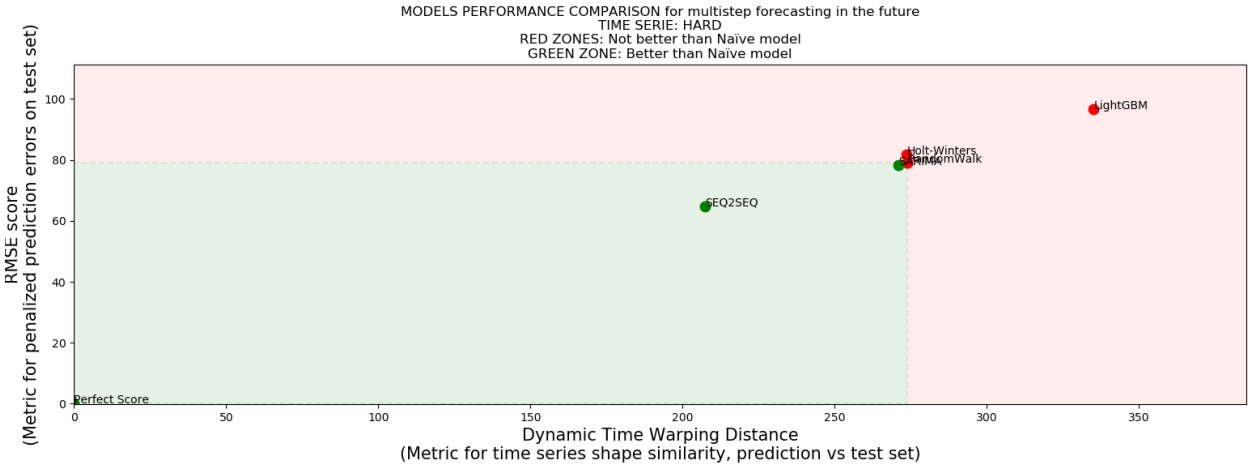
If we zoom the prediction of the best model (Seq2Seq):



Interpretation of the best model (Seq2Seq):

```
In [441]: rdr_score4.get_rdr_interpretation()
Out[441]: 'AVERAGE PERFORMANCE:With a stable trend and no major unpredictable changes,
the model is 21.14% better than a naïve random decision. The mean forecast accuracy is
62.48% (around 48.46% and 70.92% per datapoint)'
```

If we want to have more detailed information on the RdR score, we can plot the RMSE vs DTW distance like this:



The y axis is for the penalized errors while the x axis is for the shape similarity of the time series. In this, we can see that Seq2Seq was the best model while Holt-Winters and SARIMA were very near the random walk performance. LightGBM was out, the worst model in this case.

With RdR score, we can also try to get an overall score on the three datasets together:

	Dataset1 - Easy	Dataset2 - Medium	Dataset3 - Hard	Mean
LightGBM	47,24%	22,19%	-22,33%	15,70%
HoltWinter	48,10%	20,18%	-1,74%	22,18%
SARIMA	48,34%	4,77%	1,10%	18,07%
Seq2Seq	50,32%	42,79%	21,14%	38,08%

Seq2Seq win this round with mean RdR score of 38,08%, Holt-Winters in second position, SARIMA then LightGBM.

Of course, we could have tune the hyper-parameters, make more data transformations, tune neural network architecture, add exogenous data, etc. but the goal here was not to put the focus on the models. Soo please, do not make the conclusion that seq2seq is always the best model to use!

As usual, the python code of this experimental “RdR score” is avalaible on my github as jupyter notebook experiment: github.

I hope that this experiment have been useful to you!

Dave Cote, Data Scientist, M.Sc. BI