M5

THE M5 COMPETITION

Competitors' Guide

Contents

M5

Objectives

The objective of the M5 forecasting competition is to advance the theory and practice of forecasting by identifying the method(s) that provide the **most accurate point forecasts** for each of the **42,840** time series of the competition. I addition, to elicit information to estimate the **uncertainty distribution** of the realized values of these series as precisely as possible.

To that end, the participants of M5 are asked to provide **28 days ahead point forecasts (PFs)** for all the series of the competition, as well as the corresponding **median and 50%, 67%, 95%, and 99% prediction intervals (PIs)**.

The M5 differs from the previous four ones in five important ways, some of them suggested by the discussants of the M4[1] competition, as follows:

- First, it uses **grouped** unit sales data, starting at the product-store level and being aggregated to that of product departments, product categories, stores, and three geographical areas: the States of California (CA), Texas (TX), and Wisconsin (WI).
- Second, besides the time series data, it includes **explanatory variables** such as sell prices, promotions, days of the week, and special events (e.g. Super Bowl, Valentine's Day, and Orthodox Easter) that typically affect unit sales and could improve forecasting accuracy.
- Third, in addition to point forecasts, it assesses the **distribution of uncertainty**, as the participants are asked to provide information on nine indicative quantiles.
- Fourth, instead of having a single competition to estimate both the point forecasts and the uncertainty distribution, there will be **two** parallel tracks using the **same** dataset, the first requiring 28 days ahead point forecasts and the second 28 days ahead probabilistic forecasts for the median and four prediction intervals (50%, 67%, 95%, and 99%).
- Fifth, for the first time it focuses on series that display **intermittency**, i.e., sporadic demand including zeros.

Dates and hosting

The M5 will start on **March 2**, **2020** and finish on **June 30** of the same year. The competition will be run using the **Kaggle** platform. Thus, we expect many submissions from all types of forecasters including data scientists, statisticians, and practitioners, expanding the field of forecasting and eventually integrating its various approaches for improving accuracy and uncertainty estimation.

The competition will be divided into two separate Kaggle competitions, using the same dataset, with the first (**M5 Forecasting Competition – Accuracy**) requiring 28 days ahead point forecasts and the second (**M5 Forecasting Competition – Uncertainty**) 28 days ahead probabilistic forecasts for the corresponding median and four prediction intervals (50%, 67%, 95%, and 99%).

In order to support the participants to validate their forecasting approaches, the competition will include a **validation phase** that will take place from **March 2, 2020** to **31 May** of the same year. During this phase, the participants will be allowed to train their forecasting methods with the data initially provided by the organizers and validate the performance of their approaches using a hidden sample of 28 days, not made publicly available. By submitting their forecasts at the Kaggle platform (a maximum of 5 entries per day),

[1] Makridakis, S., Spiliotis, E. & Assimakopoulos, V. (2020). The M4 Competition: 100,000 time series and 61 forecasting methods. International Journal of Forecasting, 36, 54–74.

the participants will be informed about the score of their submission, which will be then published in Kaggle' s real-time leaderboard. Given this instant feedback, participants will be allowed to effectively revise and resubmit their forecasts by learning from the received feedback.

After the end of the validation phase, i.e., from **June 1, 2020** to **30 June** of the same year, the participants will be provided with the actual values of the 28 days of data used for scoring their performance during the validation phase. They will be asked then to re-estimate or adjust (if needed) their forecasting models in order to submit their final forecasts and prediction intervals for the following 28 days, i.e., the data used for the final evaluation of the participants. During this time, there will be no leaderboard, meaning that no feedback will be given to the participants about their score after submitting their forecasts. Thus, although the participants will be free to (re)submit their forecasts any time they wish (a maximum of 5 entries per day), they will not be aware of their absolute, as well as their relative performance. The **final ranks** of the participants will be made available only **at the end of competition**, when the test data will be made available. This is done in order for the competition to simulate reality as closely as possible, given that in real life forecasters do not know the future.

Note that the submission system will be open from the beginning of the competition, meaning that participants will be able to submit their final forecast from March 2, 2020 to June 30, 2020, even during the validation phase. However, as previously mentioned, the complete M5 training sample (including the 28 days used for the validation phases' leaderboard) will only become available on June 1, 2020. So, any participant submitting his/his/their final forecasts during the validation phase will be missing the last 28 days of the complete training sample.

Note also that M5 will be divided into **two tracks**, one requiring point forecasts, and one requiring the estimation of the uncertainty distribution, each with its separate prizes of $50,000. Thus, two individual competitions will be visible at the Kaggle platform, each one with its own separate leaderboard. Participants are allowed to compete and be eligible for prizes in the first track, the second track, or both.

The dataset

The M5 dataset, generously made available by **Walmart**, involves the unit sales of various products sold in the USA, organized in the form of **grouped time series**. More specifically, the dataset involves the unit sales of **3,049 products**, classified in **3 product categories** (Hobbies, Foods, and Household) and **7 product departments**, in which the above-mentioned categories are disaggregated. The products are sold across **ten stores**, located in **three States** (CA, TX, and WI). In this respect, the bottom-level of the hierarchy, i.e., product-store unit sales can be mapped across either product categories or geographical regions, as follows:

Table 1: Number of M5 series per aggregation level.

Level id	Aggregation Level	Number of series
1	Unit sales of all products, aggregated for all stores/states	1
2	Unit sales of all products, aggregated for each State	3
3	Unit sales of all products, aggregated for each store	10
4	Unit sales of all products, aggregated for each category	3
5	Unit sales of all products, aggregated for each department	7
6	Unit sales of all products, aggregated for each State and category	9

7	Unit sales of all products, aggregated for each State and department	21
8	Unit sales of all products, aggregated for each store and category	30
9	Unit sales of all products, aggregated for each store and department	70
10	Unit sales of product *x*, aggregated for all stores/states	3,049
11	Unit sales of product *x*, aggregated for each State	9,147
12	Unit sales of product *x*, aggregated for each store	30,490
Total		**42,840**

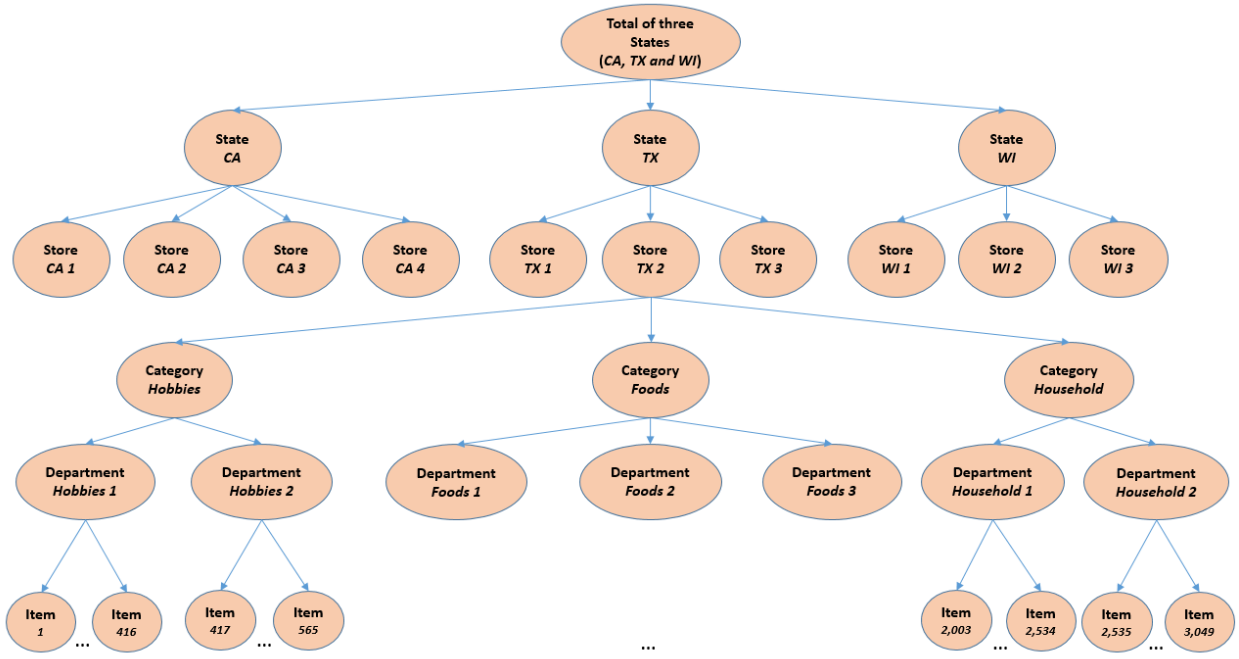

Figure 1: An overview of how the M5 series are organized.

The historical data range from **2011-01-29** to **2016-06-19**. Thus, the products have a (maximum) selling history of 1,941[2] days / 5.4 years (**test data of h=28 days not included**).

The M5 dataset consists of the following **three (3) files**:

File 1: "*calendar.csv*"
Contains information about the dates the products are sold.

- *date*: The date in a "y-m-d" format.
- *wm_yr_wk*: The id of the week the date belongs to.
- *weekday*: The type of the day (Saturday, Sunday, …, Friday).
- *wday*: The id of the weekday, starting from Saturday.
- *month*: The month of the date.
- *year*: The year of the date.

[2] This number refers to the history of the final training dataset to be provided at June 1, 2020. 28 less days will be available during the validation phase, as explained in the "dates and hosting" section.

- *event_name_1*: If the date includes an event, the name of this event.
- *event_type_1*: If the date includes an event, the type of this event.
- *event_name_2*: If the date includes a second event, the name of this event.
- *event_type_2*: If the date includes a second event, the type of this event.
- *snap_CA*, *snap_TX*, and *snap_WI*: A binary variable (0 or 1) indicating whether the stores of CA, TX or WI allow SNAP[3] purchases on the examined date. 1 indicates that SNAP purchases are allowed.

File 2: *"sell_prices.csv"*
Contains information about the price of the products sold per store and date.

- *store_id*: The id of the store where the product is sold.
- *item_id*: The id of the product.
- *wm_yr_wk*: The id of the week.
- *sell_price*: The price of the product for the given week/store. The price is provided per week (average across seven days). If not available, this means that the product was not sold during the examined week. Note that although prices are constant at weekly basis, they may change through time (both training and test set).

File 3: "*sales_train.csv*"
Contains the historical daily unit sales data per product and store.

- *item_id*: The id of the product.
- *dept_id*: The id of the department the product belongs to.
- *cat_id*: The id of the category the product belongs to.
- *store_id*: The id of the store where the product is sold.
- *state_id*: The State where the store is located.
- *d_1, d_2, …, d_i, … d_1941*: The number of units sold at day *i*, starting from 2011-01-29.

Evaluation

Forecasting horizon

The number of forecasts required, both for point and probabilistic forecasts, is **h=28 days** (4 weeks ahead).

The performance measures are **first computed for each series** separately by averaging their values across the forecasting horizon and **then averaged again across the series** in a weighted fashion (see below) to obtain the final scores.

[3] The United States federal government provides a nutrition assistance benefit called the Supplement Nutrition Assistance Program (SNAP). SNAP provides low income families and individuals with an Electronic Benefits Transfer debit card to purchase food products. In many states, the monetary benefits are dispersed to people across 10 days of the month and on each of these days 1/10 of the people will receive the benefit on their card. More information about the SNAP program can be found here: https://www.fns.usda.gov/snap/supplemental-nutrition-assistance-program

Point forecasts

The accuracy of the point forecasts will be evaluated using the **Root Mean Squared Scaled Error** (**RMSSE**), which is a variant of the well-known Mean Absolute Scaled Error (MASE) proposed by Hyndman and Koehler (2006)[4]. The measure is calculated for each series as follows:

$$\boldsymbol{RMSSE} = \sqrt{\frac{1}{h}\frac{\sum_{t=n+1}^{n+h}\left(Y_t - \widehat{Y_t}\right)^2}{\frac{1}{n-1}\sum_{t=2}^{n}(Y_t - Y_{t-1})^2}},$$

where Y_t is the actual future value of the examined time series at point *t*, $\widehat{Y_t}$ the generated forecast, *n* the length of the training sample (number of historical observations), and *h* the forecasting horizon.

Note that the denominator of RMSSE is computed only for the time-periods for which the examined product(s) are actively sold, i.e., the periods following the first non-zero demand observed for the series under evaluation.

The choice of the measure is justified as follows:

- The M5 series are characterized by intermittency, involving sporadic unit sales with lots of zeros. This means that absolute errors, which are optimized for the median, would assign lower scores (better performance) to forecasting methods that derive forecasts close to zero. However, the objective of M5 is to accurately forecast the average demand and for this reason, the accuracy measure used builds on squared errors, which are optimized for the mean.
- The measure is scale independent, meaning that it can be effectively used to compare forecasts across series with different scales.
- In contrast to other measures, it can be safely computed as it does not rely on divisions with values that could be equal or close to zero (e.g. as done in percentage errors when $Y_t = 0$ or relative errors when the error of the benchmark used for scaling is zero).
- The measure penalizes positive and negative forecast errors, as well as large and small forecasts, equally, thus being symmetric.

After estimating the RMSSE for all the 42,840 time series of the competition, the participating methods will be ranked using the **Weighted RMSSE** (**WRMSSE**), as described latter in this Guide, using the following formula:

$$\boldsymbol{WRMSSE} = \sum_{i=1}^{42{,}840} w_i * RMSSE,$$

where w_i is the weight of the i_{th} series of the competition. A lower WRMSSE score is better.

Note that the weight of each series will be computed based on the last 28 observations of the training sample of the dataset, i.e., the cumulative actual dollar sales that each series displayed in that particular

[4] R. J. Hyndman & A. B. Koehler (2006). Another look at measures of forecast accuracy. International Journal of Forecasting 22(4), 679-688.

period (sum of units sold multiplied by their respective price). An indicative example for computing the WRMSSE will be available on the GitHub[5] repository of the competition.

Probabilistic forecasts

The precision of the probabilistic forecasts will be evaluated using the **Scaled Pinball Loss** (**SPL**) function. The measure is calculated for each series and quantile as follows:

$$\mathbf{SPL(u)} = \frac{1}{h}\frac{\sum_{t=n+1}^{n+h}(Y_t - Q_t(u))u\mathbf{1}\{Q_t(u) \le Y_t\} + (Q_t(u) - Y_t)(1-u)\mathbf{1}\{Q_t(u) > Y_t\}}{\frac{1}{n-1}\sum_{t=2}^{n}|Y_t - Y_{t-1}|},$$

where Y_t is the actual future value of the examined time series at point *t*, $Q_t(u)$ the generated forecast for quantile *u*, *h* the forecasting horizon, *n* the length of the training sample (number of historical observations), and **1** the indicator function (being 1 if *Y* is within the postulated interval and 0 otherwise).

As done with RMSSE, the denominator of SPL is computed only for the time-periods for which the examined items/products are actively sold, i.e., the periods following the first non-zero demand observed for the series under evaluation.

Given that forecasters will be asked to provide the **median**, and the **50%, 67%, 95%, and 99% PIs**, u is set to u_1=0.005, u_2=0.025, u_3=0.165, u_4=0.25, u_5=0.5, u_6=0.75, u_7=0.835, u_8=0.975, and u_9=0.995. The smaller values of *u* correspond to the left side of the distribution, while the higher values to the right side of the distribution, with *u* = 0.5 being the median. The median and the 50% and 67% PIs provide a good sense of the middle of the distribution, while the 95% and 99% PIs provide information about its tails, which are important in terms of the risk of extremely high or extremely low outcomes.

After estimating the SPL for all the 42,840 time series of the competition and for all the requested quantiles, the participating methods will be ranked using the **Weighted SPL** (**WSPL**), as described latter in this Guide, divided by nine (average performance of nine quantiles across all series) , using the following formula:

$$\boldsymbol{WSPL} = \sum_{i=1}^{42{,}840} w_i * \frac{1}{9}\sum_{j=1}^{9} SPL(u_j),$$

where w_i is the weight of the i_{th} series of the competition and u_j the j_{th}out of the examined quantiles. A lower WSPL score is better.

The choice of the measure is justified as follows:

- PL is scaled in a similar fashion to that of RMSSE, meaning that it can be effectively used to compare forecasts across series with different scales. Moreover, SPL can be safely computed as it does not rely on divisions with values that could be equal to zero.
- Since M5 does not focus on a particular decision-making problem, neither defines the exact parameters of such a problem (which could also vary for different aggregation levels and series),

[5] https://github.com/Mcompetitions

it becomes evident that all quantiles could be potentially useful. Moreover, since the objective of the M5 is to estimate the uncertainty distribution of the realized values of the examined series as precisely as possible, both sides and both ends of the distribution are considered relevant. In this regard, no special weights are assigned to the examined quantiles, which are therefore equally weighted.

Note that, once again, the weight of each series will be computed based on the last 28 observations of the training sample of the dataset, i.e., the cumulative actual dollar sales that each series displayed in that particular period (sum of units sold multiplied by their respective price). An indicative example for computing the WSPL will be available on the GitHub repository of the competition.

Weighting

In contrast to the previous M competition, M5 involves the unit sales of various products of different selling volumes and prices that are organized in a hierarchical fashion. This means that, businesswise, in order for a method to perform well, it must provide accurate forecasts across all hierarchical levels, especially for series of high importance, i.e. for series that represent significant sales, measured in US dollars. In other words, we expect from the best performing forecasting methods to derive lower forecasting errors for the series that are more valuable for the company.

To that end, the forecasting errors computed for each participating method (both RMSSE and SPL) will be weighted across the M5 series based on their **cumulative actual dollar sales**, which is a good and objective proxy of their actual value for the company in monetary terms. The cumulative dollar sales will be computed using **the last 28 observations of the training sample** (sum of units sold multiplied by their respective price), i.e., a period equal to the forecasting horizon. Note that since both the number of units being sold and their respective price change through time, this estimation is based on the sum of the corresponding daily dollar sales.

Below you may find a simple, yet indicative example of how these weights will be computed:

Assume that two products of the same department, A and B, are sold in a store at WI and we are interested in forecasting the unit sales of these two products, as well as their aggregate sales. Thus, in this example, we consider two different aggregation levels (K=2), the first level consisting of two series (series A and B) and the second level of a single series (sum of series A and B).

Product A displayed a total of $10 in sales in the last 28 days of the training sample, while product B $12. Thus, the aggregate dollar sales of products A and B in the last 28 days were $22. Assume also that a forecasting method was used to derive point forecasts for product A, product B, and their aggregate unit sales, displaying errors $RMSSE_A$=0.8, $RMSSE_B$=0.7, and RMSSE=0.77, respectively. If the M5 dataset involved just those three series, the final WRMSSE score of the method would be

$$WRMSSE = RMSSE_A * w_1 + RMSSE_B * w_2 + RMSSE * w_3 =$$

$$RMSSE_A * \frac{1}{K} * \frac{\$\,Sales_A}{\$\,Sales_A + \$\,Sales_B} + RMSSE_B * \frac{1}{K} * \frac{\$\,Sales_B}{\$\,Sales_A + \$\,Sales_B} + RMSSE * \frac{1}{K} * \frac{\$\,Sales}{\$\,Sales_A + \$\,Sales_B} =$$

$$0.8 * \frac{1}{2} * \frac{10}{10+12} + 0.7 * \frac{1}{2} * \frac{12}{10+12} + 0.77 * \frac{1}{2} * 1 = 0.758.$$

This weighting scheme can be expanded in order to consider more stores, geographical regions, product categories, and product departments, as previously described. Since the M5 competition involves twelve aggregation levels, K is set equal to 12, with the weights of the series being computed so that they sum to one at each aggregation level.

Respectively, RMSSE, which is used in the above equation for estimating WRMSSE, can be replaced with SPL to compute WSPL.

Note that **all hierarchical levels are equally weighted**. The reason is because the total dollar sales of a product, measured across all three States, are equal to the sum of the dollar sales of this product when measured across all ten stores. Similarly, because the total dollar sales of a product category of a store are equal to the sum of the dollar sales of the departments that this category consists of, as well as the sum of the dollar sales of the products of the corresponding departments. Moreover, as previously discussed for the case of the probabilistic forecasts, M5 does not focus on a particular decision-making problem, which means that there is no reason for weighting unequally the individual levels of the hierarchy.

An indicative example for computing WRMSSE and WSPL will be available on the GitHub repository of the competition, indicating among others the exact weight of each series in the competition.

The Prizes

Distribution of prize money

There will be **12** major prizes awarded to the winners of the M5 Competition, which will be further distributed among the participants based on (i) the hierarchical levels that their forecasts exceled and (ii) the quantiles of the uncertainty distribution that were better captured. The Prizes will be awarded on **December 8, 2020**, during the **M5 Forecasting Conference** to be held in New York City. At this date, Kaggle will be issuing the payments digitally using its collaborating firm Payoneer.

The total of the $100,000 prize money will be distributed equally between the Forecasting and Uncertainty M5 competition as follows:

Prize id	Prize	Amount
1A	Most accurate point forecasts	$25,000
2A	Second most accurate point forecasts	$10,000
3A	Third most accurate point forecasts	$5,000
4A	Fourth most accurate point forecasts	$3,000
5A	Fifth most accurate point forecasts	$2,000
6A	Most accurate student point forecasts	$5,000
	Total: M5 Forecasting Competition - Point Forecasts	**$50,000**
1B	Most precise estimation of the uncertainty distribution	$25,000
2B	Second most precise estimation of the uncertainty distribution	$10,000
3B	Third most precise estimation of the uncertainty distribution	$5,000
4B	Fourth most precise estimation of the uncertainty distribution	$3,000
5B	Fifth most precise estimation of the uncertainty distribution	$2,000
6B	Most precise student estimation of the uncertainty distribution	$5,000

	Total: M5 Forecasting Competition - Uncertainty Distribution	**$50,000**
Total: M5 Competition		**$100,000**

Reproducibility

The prerequisite for winning any prize will be that the code used for generating the forecasts, with the exception of companies providing forecasting services and those claiming proprietary software, will be put on GitHub, **not later than 14 days after the end of the competition** (i.e., the 14th of July, 2020). In addition, there must be instructions on how to exactly reproduce the M5 submitted forecasts. In this regard, individuals and companies will be able to use the code and the instructions provided, crediting the person/group that has developed them, to improve their organizational forecasts.

Companies providing forecasting services and those claiming proprietary software will have to provide the organizers with a detailed description of how their forecasts were made and a source, or execution file for reproducing their forecasts. Given the critical importance of objectivity and replicability, such description and file will be mandatory for winning any prize of the competition. An execution file can be submitted in case that the source program needs to be kept confidential, or, alternatively, a source program with a termination date for running it.

After receiving the code/program/files for reproducing the submitted forecasts, the organizers will evaluate their results in terms of reproducibility. Since some methods may involve random initializations, any method that displays a replicability rate higher than 98% will be considered as fully replicable and be awarded the prize, exactly as done in M4. Otherwise, the prize will be given to the next best-performing and fully reproducible submission.

Publications

Similar to the M3 and M4 competitions, there will be a special issue of the **International Journal of Forecasting** (**IJF**) exclusively devoted to all aspects of the M5 Competition with special emphasis on what we have learned and how we can use such learning to improve the theory and practice of forecasting as well as expand its usefulness and applicability.

The Benchmarks

Like done in the M4 competition, there will be benchmark methods, twenty-four (24) for point forecasts, and six (6) for probabilistic ones. As these methods are well known, readily available, and straightforward to apply, the accuracy of the new ones submitted to the M5 Competition must provide superior accuracy in order to be considered and used in practice (taking also into account the computational time it would be required to utilize a more accurate method versus the benchmarks whose computational requirements are minimal).

Point forecasts

Statistical Benchmarks

1. Naive: A random walk model, defined as

$$\hat{Y}_{n+i} = Y_n \, , i = 1, 2, \dots h \, .$$

The forecasting method is used for predicting the series of the lowest level of the hierarchy (level 12 of Table 1) and the bottom-up method is then used for reconciliation.

2. Seasonal Naive (sNaive): Like Naive, but this time the forecasts of the model are equal to the last known observation of the same period in order for it to capture possible weekly seasonal variations. The forecasting method is used for predicting the series of the lowest level of the hierarchy (level 12 of Table 1) and the bottom-up method is then used for reconciliation.

3. Simple Exponential Smoothing[6] (SES): The simplest exponential smoothing model, aimed at predicting series without a trend, defined as

$$\hat{Y}_t = aY_t + (1-a)\hat{Y}_{t-1}.$$

The smoothing parameter *a* is selected from the range [0.1, 0.3] by minimizing the insample mean squared error (MSE) of the model, while the first observation of the series is used for initialization. The forecasting method is used for predicting the series of the lowest level of the hierarchy (level 12 of Table 1) and the bottom-up method is then used for reconciliation.

4. Moving Averages (MA): Forecasts are computed by averaging the last *k* observations of the series, as follows

$$\hat{Y}_t = \frac{\sum_{i=1}^{k} Y_{t-i}}{k},$$

where *k* is selected from the range [2, 5] by minimizing the insample MSE. The forecasting method is used for predicting the series of the lowest level of the hierarchy (level 12 of Table 1) and the bottom-up method is then used for reconciliation.

5. Croston's method[7] (CRO): The method proposed by Croston to forecast series that display intermittent demand. The method decomposes the original series into the non-zero demand size z_t and the inter-demand intervals p_t, deriving forecasts as follows:

$$\hat{Y}_t = \frac{\hat{z}_t}{\hat{p}_t},$$

where both z_t and p_t are predicted using SES. The smoothing parameter of both components is set equal to 0.1. The first observation of the components are used for initialization. The forecasting method is used for predicting the series of the lowest level of the hierarchy (level 12 of Table 1) and the bottom-up method is then used for reconciliation.

6. Optimized Croston's method (optCro): Like CRO, but this time the smoothing parameter is selected from the range [0.1, 0.3], like done with SES, in order to allow for more flexibility. The non-zero demand size and the inter-demand intervals are smoothed separately using (potentially) different *a* parameters. The forecasting method is used for predicting the series of the lowest level of the hierarchy (level 12 of Table 1) and the bottom-up method is then used for reconciliation.

[6] Gardner Jr., E. S. (1985). Exponential smoothing: The state of the art. Journal of Forecasting, 4, 1–28.
[7] Croston, J. D. (1972). Forecasting and stock control for intermittent demands. Journal of the Operational Research Society, 23, 289–303.

7. Syntetos-Boylan Approximation[8] (SBA): A variant of the Croston's method that utilizes a debiasing factor as follows:

$$\hat{Y}_t = 0.95\frac{\hat{z}_t}{\hat{p}_t}$$

The forecasting method is used for predicting the series of the lowest level of the hierarchy (level 12 of Table 1) and the bottom-up method is then used for reconciliation.

8. Teunter-Syntetos-Babai method[9] (TSB): A modification to Croston's method that replaces the inter-demand intervals component with the demand probability, d_t, being 1 if demand occurs at time t and 0 otherwise. Similarly to Croston's method, d_t is forecasted using SES. The smoothing parameters of d_t and z_t may differ, exactly as optCRO. The forecast is given as follows:

$$\hat{Y}_t = \hat{d}_t\hat{z}_t$$

The forecasting method is used for predicting the series of the lowest level of the hierarchy (level 12 of Table 1) and the bottom-up method is then used for reconciliation.

9. Aggregate-Disaggregate Intermittent Demand Approach[10] (ADIDA): Temporal aggregation is used for reducing the presence of zero observations, thus mitigating the undesirable effect of the variance observed in the intervals. ADIDA uses equally sized time buckets to perform non-overlapping temporal aggregation and predict the demand over a pre-specified lead-time. The time bucket is set equal to the mean inter-demand interval. SES is used to obtain the forecasts. The forecasting method is used for predicting the series of the lowest level of the hierarchy (level 12 of Table 1) and the bottom-up method is then used for reconciliation.

10. Intermittent Multiple Aggregation Prediction Algorithm[11] (iMAPA): Another way for implementing temporal aggregation in demand forecasting. However, in contrast to ADIDA that considers a single aggregation level, iMAPA considers multiple ones, aiming at capturing different dynamics of the data. Thus, iMAPA proceeds by averaging the derived point forecasts, generated using SES. The maximum aggregation level is set equal to the maximum inter-demand interval. The forecasting method is used for predicting the series of the lowest level of the hierarchy (level 12 of Table 1) and the bottom-up method is then used for reconciliation.

11. Exponential Smoothing[12] - Top-Down (ES_td): An algorithm is used to select the most appropriate exponential smoothing model for predicting the top-level series of the hierarchy (level 1 of Table 1),

[8] Syntetos, A. A. & Boylan, J. E. (2005). The accuracy of intermittent demand estimates. International Journal of Forecasting, 21, 303–314.

[9] Teunter, R. H., Syntetos, A. A. & Babai, M. Z. (2011). Intermittent demand: Linking forecasting to inventory obsolescence. European Journal of Operational Research, 214, 606–615.

[10] Nikolopoulos, K., Syntetos, A. A., Boylan, J. E., Petropoulos, F. & Assimakopoulos, V. (2011). An aggregate-disaggregate intermittent demand approach (ADIDA) to forecasting: an empirical proposition and analysis. Journal of the Operational Research Society, 62, 544–554.

[11] Petropoulos, F. & Kourentzes, N. (2015). Forecast combinations for intermittent demand. Journal of the Operational Research Society, 66, 914–924

[12] Hyndman, R.J., Koehler, A.B., Snyder, R.D. & Grose, S. (2002). A state space framework for automatic forecasting using exponential smoothing methods. International Journal of Forecasting, 18 (3), 439-454.

indicated through information criteria. The top-down method is used for reconciliation (based on historical proportions, estimated for the last 28 days).

12. Exponential Smoothing – Bottom-Up (ES_bu): An algorithm is used to select the most appropriate exponential smoothing model for predicting the bottom-level series of the hierarchy (level 12 of Table 1), indicated through information criteria. The forecasting method is used for predicting the series of the lowest level of the hierarchy (level 12 of Table 1) and the bottom-up method is then used for reconciliation.

13. Exponential Smoothing with eXplanatory variables (ESX): Similar to ES, but this time two explanatory variables are used as regressors to improve forecasting accuracy by providing additional information about the future. The first variable is discrete and takes values 0, 1, 2 or 3, based on the number of States that allow SNAP purchases on the examined date. The second variable is binary and indicates whether the examined date includes a special event (1) or not (0). The top-down method is used for reconciliation (based on historical proportions, estimated for the last 28 days).

14. AutoRegressive Integrated Moving Average[13] - Top-Down (ARIMA_td): An algorithm is used to select the most appropriate ARIMA model for predicting the top-level series of the hierarchy (level 1 of Table 1), indicated through information criteria. The top-down method is used for reconciliation (based on historical proportions, estimated for the last 28 days).

15. AutoRegressive Integrated Moving Average – Bottom-Up (ARIMA_bu): An algorithm is used to select the most appropriate ARIMA model for predicting the bottom-level series of the hierarchy (level 12 of Table 1), indicated through information criteria. The forecasting method is used for predicting the series of the lowest level of the hierarchy (level 12 of Table 1) and the bottom-up method is then used for reconciliation.

16. AutoRegressive Integrated Moving Average with eXplanatory variables (ARIMAX): Similar to ARIMA, but this time two external variables are used as regressors to improve forecasting accuracy by providing additional information about the future, exactly as done for the case of ESX. The top-down method is used for reconciliation (based on historical proportions, estimated for the last 28 days).

Machine Learning Benchmarks

17. Multi-Layer Perceptron (MLP): A single hidden layer NN of 14 input nodes (last two weeks of available data), 28 hidden nodes, and one output node. The Scaled Conjugate Gradient method is used for estimating the weights that are initialized randomly, while the maximum iterations are set equal to 500. The activation functions of the hidden and output layers are the logistic and linear one, respectively. In total, 10 MLPs are trained to forecast each series and then the median operator is used to average the individual forecasts in order to mitigate possible variations due to poor weight initializations. The forecasting method is used for predicting the series of the lowest level of the hierarchy (level 12 of Table 1) and the bottom-up method is then used for reconciliation.

18. Random Forest (RF): This is a combination of multiple regression trees, each one depending on the values of a random vector sampled independently and with the same distribution. Given that RF averages

[13] Hyndman, R. & Khandakar Y. (2008). Automatic time series forecasting: the forecast package for R. Journal of Statistical Software, 26, 1-22.

the predictions of multiple trees, it is more robust to noise and less likely to over-fit on the training data. We consider a total of 500 non-pruned trees and four randomly sampled variables at each split. Bootstrap sampling is done with replacement. Like done in MLP, the last 14 observations of the series are considered for training the model. The forecasting method is used for predicting the series of the lowest level of the hierarchy (level 12 of Table 1) and the bottom-up method is then used for reconciliation.

19. Global Multi-Layer Perceptron (GMLP): Like MLP, but this time, instead of training multiple models, one for each series, a single model that learns across all series is constructed. This is done given that M4 indicated the beneficial effect of cross learning. The last 14 observations of each series are used as inputs, along with information about the coefficient of variation of non-zero demands (CV^2) and the average number of time-periods between two successive non-zero demands (ADI). This additional information is used in order to facilitate learning across series of different characteristics. The forecasting method is used for predicting the series of the lowest level of the hierarchy (level 12 of Table 1) and the bottom-up method is then used for reconciliation.

20. Global Random Forest (GRF): Like GMLP, but instead of using an MLP for obtaining the forecasts, a RF is exploited instead. The forecasting method is used for predicting the series of the lowest level of the hierarchy (level 12 of Table 1) and the bottom-up method is then used for reconciliation.

Combination Benchmarks

21. Average of ES and ARIMA, as computed using the bottom-up approach (Com_b): The simple arithmetic mean of ES_bu and ARIMA_bu.

22. Average of ES and ARIMA, as computed using the top-down approach (Com_t): The simple arithmetic mean of ES_td and ARIMA_td.

23. Average of the two ES methods, the first computed using the top-down approach and the second using the bottom-up approach (Com_tb): The simple arithmetic mean of ES_td and ES_bu.

24. Average of the global and local MLPs (Com_lg): The simple arithmetic mean of MLP and GMLP. The bottom-up method is then used for reconciliation.

Observe that the benchmark methods {1-10, 12, 15, 17-20} are applied at the product-store level of the hierarchically structured dataset. Thus, the bottom-up method is used for obtaining reconciled forecasts for the rest of the hierarchical levels. On the other hand, the benchmark methods {11, 13, 14, 16} are applied at the top level of the hierarchically structured dataset. Thus, the top-down method is used for obtaining reconciled forecasts for the rest of the hierarchical levels (based on historical proportions, estimated for the last 28 days).

Probabilistic forecasts

i. Naive: Similar implementation to the Naive 1 used for computing point forecasts.

ii. Seasonal Naive (sNaive): Similar implementation to the sNaive one used for computing point forecasts.

iii. Simple Exponential Smoothing (SES): Similar implementation to the SES one used for computing point forecasts.

iv. Exponential Smoothing (ES): Similar implementation to the ES_bu one used for computing point forecasts.

v. AutoRegressive Integrated Moving Average (ARIMA): Similar implementation to the ARIMA_bu one used for computing point forecasts.

vi. Kerner density estimate (Kernel): A kernel is used to estimate the corresponding quantiles in the historical data that are then used as probabilistic forecasts.

The code for generating the forecasts of the abovementioned benchmarks will be available on the GitHub repository of the competition.

Benchmarks are not eligible for a prize, meaning that the total amount will be distributed among the competitors even if the benchmarks perform better than the forecasts submitted by the participants. Similarly, any participating method associated with the organizers and the data provider, will not be eligible for a price.

Submission

The forecasts for both competitions will be submitted through the Kaggle platform. The templates provided by the organizers though the platform can be used for this purpose.

Note that the template of the point forecasts (M5 Forecasting - Accuracy) refers only to the 30,490 series that consist the lowest hierarchical level of the dataset (level 12 of Table 1) and not all 42,840 of the competition (all levels of Table 1). This is done because M5, in contrast to M4, M3, and other forecasting competitions where time series are mostly unrelated, deals among others with a real-life hierarchical forecasting problem. This means that the submitted forecasts must follow this hierarchical concept and, as a result, be coherent (forecasts at the lower levels have to sum up to the ones of the higher levels). In other words, it is assumed that the forecasting approach used for forecasting all 42,840 series of the competition derived coherent forecasts and, therefore, the forecasts of all levels can be automatically computed by aggregating (summing) the ones of the lowest level of the hierarchy.

It is important to note that the participants are completely free to use the forecasting approaches of their choice for forecasting the individual series. However, having done that, and by submitting just the forecasts of the lowest level, it will be assumed that the derived forecasts were reconciled before submitted for the final evaluation. For instance, a participant may forecast just the series at the bottom-level and derive the remaining forecasts using the bottom-up reconciliation method. Another participant may forecast just the series at the top level and get the ones at the lower levels using proportions (top-down reconciliation method). A mix of the previous two approaches is also possible (middle-out reconciliation method). Finally, predicting the series of all levels and getting the ones of the lowest level through an appropriate weighting scheme is also an option. The benchmarks describe some of these options, involving some indicative forecasting approaches that utilize the bottom-up (e.g. benchmark #12) and the top-down (e.g. benchmark #11) reconciliation method, as well as the combination of these two (e.g. benchmark #23).

Finally, given that there is not a direct and well-established way for reconciling probabilistic forecasts, the template of the probabilistic forecasts (M5 Forecasting - Uncertainty) requires inputting all 42,840 series

of the competition. Thus, in this case, participants do not need to reconcile the forecasts using any of the above-mentioned approaches.