M5

THE M5 COMPETITION

Competitors' Guide

Contents

UNIVERSITY of NICOSIA
FORECASTING & STRATEGY UNIT National Technical University of Athens
INSEAD The Business School for the World®

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 **43,204** time series of the competition, as well as the methods that elicit information to estimate the uncertainty distribution of these forecasts as precisely as possible.

To that end, the participants of M5 are asked to provide **28 days ahead point forecasts** for all the series of the competition, as well as the corresponding **50th, 80th, 90th, and 95th percentiles**.

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

- First, it uses **hierarchical** 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, day of the week, and special events (e.g. Super Bowl, Valentine's Day, and Orthodox Easter) that typically affect sales and could be used to improve forecasting accuracy.
- Third, in addition to point forecasts, it assesses the **distribution of uncertainty**, as the participants are asked to provide information on four indicative percentiles.
- Fourth, 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 M5 dataset will become publicly available on the starting date of the competition.

The competition will be run using the ***Kaggle*** platform. Thus, we expect a lot of submissions, including forecasters of both statistical and machine learning background, expanding that way the field of forecasting and integrating its various approaches for improving accuracy and uncertainty estimation.

Note that in contrast to what is typically done in Kaggle competitions, M5 will not involve a real-time leaderboard. This means that the participants will be free to (re)submit their forecasts on daily basis but will not be aware of their absolute, as well as their relative performances. The ranks of the participating methods will be made available only at the end of the competition, when the organizers will have made publicly available the test sample of the dataset by sharing it with Kaggle. This is done in order for the competition to simulate reality as closely as possible, keeping in mind that in real life forecasters know little about the future.

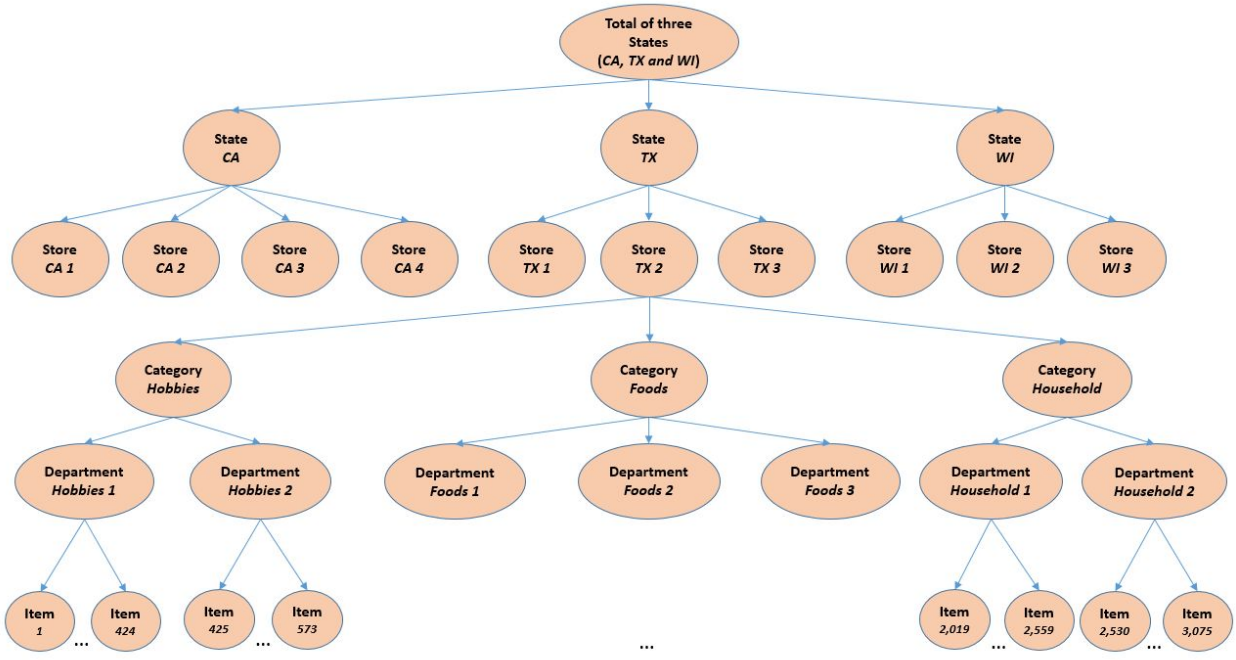
The dataset

The M5 dataset, generously made available by **Walmart**, involves the sales of various products sold in the USA, organized in the form of **grouped time series**. More specifically, the dataset involves the sales of **3,075 products**, classified in **3 product categories** (Hobbies, Foods, and Household) and **7 product**

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

departments, in which the above-mentioned categories are disaggregated. The products are sold across **10 stores**, located in **3 States** (CA, TX, and WI). In this respect, the bottom-level of the hierarchy, i.e., product-store sales, can be mapped either across product categories or geographical regions, as follows:

Aggregation Level	Number of series
Sales of all products, aggregated for all stores/states	1
Sales of all products, aggregated for each State	3
Sales of all products, aggregated for each store	10
Sales of all products, aggregated for each category	3
Sales of all products, aggregated for each department	7
Sales of all products, aggregated for each State and category	9
Sales of all products, aggregated for each State and department	21
Sales of all products, aggregated for each store and category	30
Sales of all products, aggregated for each store and department	70
Sales of product *x*, aggregated for all stores/states	3,075
Sales of product *x*, aggregated for each State	9,225
Sales of product *x*, aggregated for each store	30,750
Total	**43,204**



The historical data range from **2011-01-29** to **2016-06-19**. Thus, the products have a (maximum) selling history of 1,941 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.
- *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[2] 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. Price is provided per week (average across seven days). If not available, this means that the product was not sold during the examined week.

File 3: "*sales_train.csv*"
Contains the historical daily 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 department 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 products 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).

[2] 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

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.

Weighting

In contrast to the previous M competition, M5 involves the sales of various products 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 **aggregate 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 of more value for the company. To that end, the forecasting errors computed for each participating method will be weighted across the M5 series based on the aggregate sales that each series represents, i.e. a proxy of their actual value for the company in monetary terms.

Assume that two products of the same department, A and B, are sold in a store at WI. Product A, of price \$1, displays 10 sales in the **testing period**, while product B, of price \$2, displays 6 sales. The aggregate sales of product A will be \$1*10=\$10, while the aggregate sales of product B will be \$2*6=\$12. Assume also that a forecasting method was used to forecast the sales of product A, product B, and their aggregate sales, displaying errors E_A, E_B, and E, respectively. If the M5 dataset involved just those three series, the final score of the method would be $\frac{1}{2}(E_A * \frac{10}{10+12} + E_B * \frac{12}{10+12} + E)$.

This weighting scheme can be expanded in order to consider more stores, geographical regions, product categories, and product departments, as previously described. Note that, based on the considered scheme, **all hierarchical levels are equally weighted**. This is because the total sales of a product, measured across all three States, are equal to the sum of the sales of this product when measured across all ten stores, or similarly, because the total sales of a product category of a store are equal to the sum of the sales of the departments that this category consists of, as well as the sum of the sales of the products of the corresponding departments.

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 Absolut Scaled Error (MASE) proposed by Hyndman and Koehler (2006)[3]. The measure is calculated as follows:

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

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

The choice of the measure is justified as follows:

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

- The M5 series are characterized by intermittency, involving lots of zeros. This means that absolute errors, which are optimized for the median, would assign lower scores to forecasting methods that derive forecasts close to zero. However, the objective of M5 is to accurately forecast the average demand. Thus, the accuracy measure used depends 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 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 43,204 time series of the competition, the participating methods will be ranked using the **Weighted RMSSE** (**WRMSSE**), as described earlier. Once again, note that the weight of each series will be computed based on the test sample of the dataset, i.e., future sales and prices.

An indicative example for computing the WRMSSE will be available on the GitHub repository of the competition.

Probabilistic forecasts

The precision of the probabilistic forecasts will be evaluated using the **Scaled Pinball Loss** (**SPL**) function, as follows:

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

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** is the indicator function (being 1 if *Y* is within the postulated interval and 0 otherwise). Given that forecasters will be asked to provide the **50th, 80th, 90th, and 95th percentiles**, u is set to 0.5, 0.2, 0.1 and 0.05, respectively.

After estimating the SPL for all the 43,204 time series of the competition and for all the requested percentiles, the participating methods will be ranked using the **Weighted SPL** (**WSPL**), as described earlier, divided by four (average performance of four percentiles across all series). Once again, note that the weight of each series will be computed based on the test sample of the dataset, i.e., future sales and prices.

An indicative example for computing the WSPL will be available on the GitHub repository of the competition.

Reproducibility

The prerequisite for the Full Reproducibility Prizes 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 10 days after the end of the competition (i.e., the 10th 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 participating in 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.

The Prizes

There will be **10** Prizes awarded to the winners of the M5 Competition. The exact cash amounts to be granted (at present standing at $100,000, with the expectation to be raised to $150,000) will depend on securing additional sponsors and will be announced later. Proportionally, the total amount granted will be distributed as follows:

There will be a total of $100,000 in prize money allocated as follows:

Most accurate point forecast	$25,000
Second most accurate point forecast	$10,000
Third most accurate point forecast	$5,000
Most precise estimation of the uncertainty distribution	$25,000
Second most precise estimation of the uncertainty distribution	$10,000
Third most precise estimation of the uncertainty distribution	$5,000
Most accurate student point forecast	$5,000
Most precise student estimation of the uncertainty distribution	$5,000
20 Special prizes to invited companies winning some part of the most accurate forecast or the most precise estimation of the uncertainty distribution	$10,000
Total:	$100,000

The amounts gathered for the M5 Prizes have already exceeded $100K and it is expected to reach the 150,000 mark in which case the amount listed above will be increased and new prizes will be added.

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 field of forecasting and expand its usefulness and applicability.

The Benchmarks

Like done in the M4 competition, there will be fourteen (14) benchmark methods, ten (10) statistical and four (4) Machine Learning (ML) ones. As these methods are well known, readily available, and straightforward to apply, the accuracy of the new ones proposed in the M4 Competition must provide superior accuracy in order to be adopted 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).

Statistical Benchmarks

1. Naive: A random walk model, defined as

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

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.

3. Simple Exponential Smoothing[4] (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.

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.

[4] Gardner Jr., E. S. (1985). Exponential smoothing: The state of the art. Journal of Forecasting, 4, 1–28.

5. Croston's method[5] (CRO): The method proposed by Croston to forecast series that display intermittent demand. Th ε 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.

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.

7. Syntetos-Boylan Approximation[6] (SBA): A variant of the Croston's method that utilizes a debiasing factor as follows:

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

8. Teunter-Syntetos-Babai method[7] (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:

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

9. Aggregate-Disaggregate Intermittent Demand Approach[8] (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.

10. Intermittent Multiple Aggregation Prediction Algorithm[9] (iMAPA): Another way for implementing temporal aggregation in demand forecasting. However, in contrast to ADIDA which 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 at each temporal level, generated using SES. The maximum aggregation level is set equal to the maximum inter-demand interval.

---

[5] Croston, J. D. (1972). Forecasting and stock control for intermittent demands. Journal of the Operational Research Society, 23, 289–303.

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

[7] 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.

[8] 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.

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

Machine Learning Benchmarks

11. 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 which 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.

12. 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 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 4 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.

13. 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.

14. Global Random Forest (GRF): Like GMLP, but instead of using an MLP for obtaining the forecasts, a RF is exploited instead.

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

Note that the benchmark methods are applied at the product-store level of the hierarchically structured dataset. The bottom-up method is then used for obtaining reconciled forecasts at the rest of the hierarchical levels.

Also note that 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.