Capstone Project Report

Kaggle competition: M5 Forecasting - Accuracy

Machine Learning Engineer Nanodegree

Olga Moreva

09.04.2020

I. DEFINITION

PROJECT OVERVIEW

'Walmart Inc. is an American multinational retail corporation that operates a chain of hypermarkets, discount department stores, and grocery stores'[1]. The goal of this competition is to predict 'the unit sales of various products sold in the USA by Walmart'[2]. A good forecasting model helps retailers to plan marketing companies, manage cash flow, manage their inventory and optimize their supply chain leading to increase of profit, whereas 'inaccurate business forecasts could result in actual or opportunity losses.' Forecasting is an active area of research, in this project I concentrate on quantitative methods.

PROBLEM STATEMENT

The goal of this competition is to 'to forecast daily sales for the next 28 days' based on the data from 'stores in three US States (California, Texas, and Wisconsin)' . The data 'includes item level, department, product categories, and store details. In addition, it has explanatory variables such as price, promotions, day of the week, and special events. Together, this robust dataset can be used to improve forecasting accuracy.'

EVALUATION METRIC

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)[3]. The measure is calculated for each series as follows:

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

where Y_t is the actual future value of the examined time series at point t, $\hat{Y}_t$ the generated forecast, n the length of the training sample (number of historical observations), and h the forecasting horizon.[4]

[1] https://en.wikipedia.org/wiki/Walmart

[2] All citations in this document if not stated overwise are from the M5 Forecasting - Accuracy challenge https://www.kaggle.com/c/m5-forecasting-accuracy/overview/description

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

[4] THE M5 COMPETITION Competitors' Guide, https://mofc.unic.ac.cy/m5-competition/

II. ANALYSIS

DATA EXPLORATION

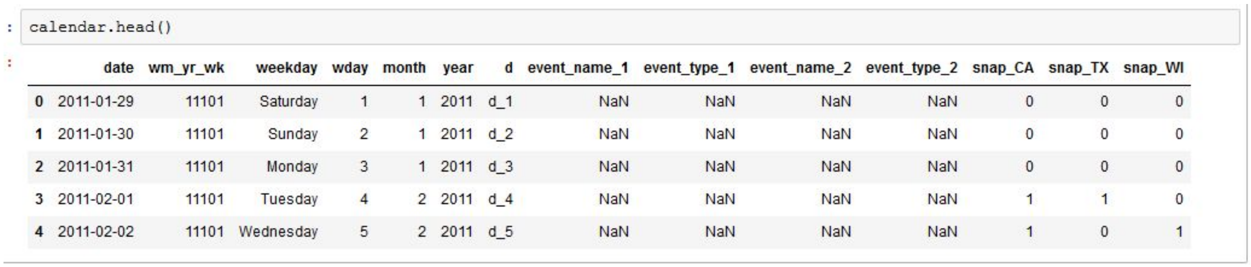
Section Dataset of the competition guidelines[5] provides an excellent description of the data and we present it here. The text in this subsection (till Size of data) is taken from the section Dataset of the competition guidelines.

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).

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.



```
calendar.head()
```

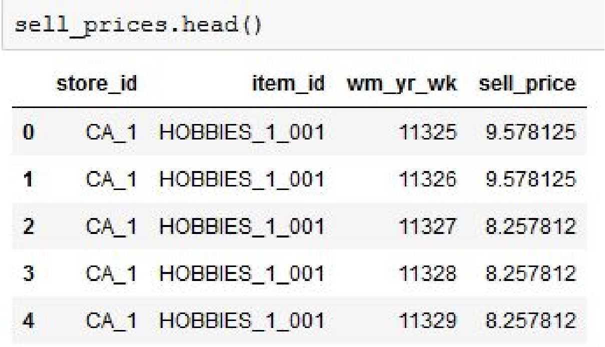
	date	wm_yr_wk	weekday	wday	month	year	d	event_name_1	event_type_1	event_name_2	event_type_2	snap_CA	snap_TX	snap_WI
0	2011-01-29	11101	Saturday	1	1	2011	d_1	NaN	NaN	NaN	NaN	0	0	0
1	2011-01-30	11101	Sunday	2	1	2011	d_2	NaN	NaN	NaN	NaN	0	0	0
2	2011-01-31	11101	Monday	3	1	2011	d_3	NaN	NaN	NaN	NaN	0	0	0
3	2011-02-01	11101	Tuesday	4	2	2011	d_4	NaN	NaN	NaN	NaN	1	1	0
4	2011-02-02	11101	Wednesday	5	2	2011	d_5	NaN	NaN	NaN	NaN	1	0	1

- *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.
- d: the id of the day
- *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.

[5] THE M5 COMPETITION Competitors' Guide, https://mofc.unic.ac.cy/m5-competition/

- *snap_CA*, *snap_TX*, and *snap_WI*: A binary variable (0 or 1) indicating whether the stores of CA, TX or WI allow SNAP [6]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.

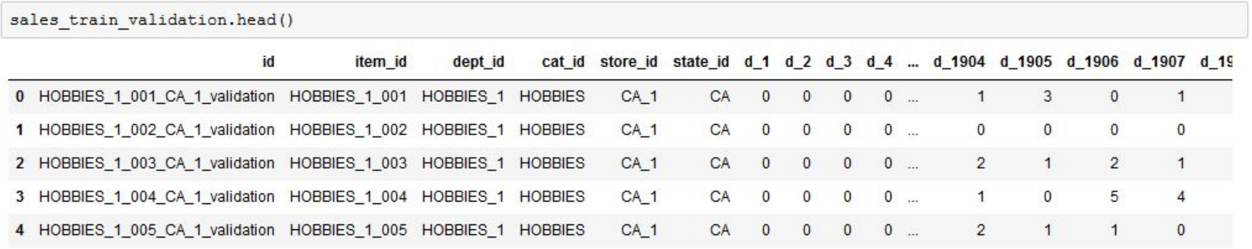


```
sell_prices.head()
```

	store_id	item_id	wm_yr_wk	sell_price
0	CA_1	HOBBIES_1_001	11325	9.578125
1	CA_1	HOBBIES_1_001	11326	9.578125
2	CA_1	HOBBIES_1_001	11327	8.257812
3	CA_1	HOBBIES_1_001	11328	8.257812
4	CA_1	HOBBIES_1_001	11329	8.257812

- *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_validation.csv*" contains the historical daily unit sales data per product and store.



```
sales_train_validation.head()
```

	id	item_id	dept_id	cat_id	store_id	state_id	d_1	d_2	d_3	d_4	...	d_1904	d_1905	d_1906	d_1907	d_19
0	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	1	3	0	1	
1	HOBBIES_1_002_CA_1_validation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	0	0	0	0	
2	HOBBIES_1_003_CA_1_validation	HOBBIES_1_003	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	2	1	2	1	
3	HOBBIES_1_004_CA_1_validation	HOBBIES_1_004	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	1	0	5	4	
4	HOBBIES_1_005_CA_1_validation	HOBBIES_1_005	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	2	1	1	0	

[6] 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

- *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.

SIZE OF DATA

1. calendar.csv has 1969 rows and 14 columns.
2. sales_train_validation.csv has 30490 rows and 1919 columns.
3. sample_submission.csv has 60980 rows and 29 columns.
4. sell_prices.csv has 6841121 rows and 4 columns.

File ***"sales_train_validation.csv"*** contains observations in the wide format. We will work with data in long format, the reshaped dataset consists of 58 327 370 rows and 8 columns occupies 3.2 GB of memory even without additional features.

FEATURE ENGINEERING

There is no NaN values in the datasets.

The initial feature matrix X has the categorical features 'id', 'item_id', 'dept_id', 'cat_id', 'store_id', and 'state_id', string feature 'd' and the target variable 'demand'.

START OF SALES

The dataset contains demand on the dates before product release, i.e. all zeros. For each 'id' we determine the start of the sales date (feature 'start_date') and remove all rows before this date.

CALENDAR FEATURES

We add the following features from ***"calendar.csv":***
'date', 'wm_yr_wk', 'weekday', 'year', 'month', 'event_name_1', 'event_type_1', 'event_name_2', 'event_type_2', 'snap_CA', 'snap_TX', 'snap_WI', 'd'

TREND

We add the number of days from the beginning of sales and from the first date in the dataset. These variables can be used to model trends in demand.

SALES PATTERN

In this part we determine forecastability of the time series, as it was done at https://github.com/Mcompetitions/M5-methods. The demand classification proposed by 2005, Syntetos, Boylan, and Croston[7] defines the following demand categories

- Smooth demand: regular demand over time with a limited vari- ation in quantity
- Intermittent demand: extremely sporadic demand, with no accentuated variability in the quantity of the single demand;
- Erratic demand: regular distribution over time, but large variation in quantity;
- Lumpy demand: extremely sporadic demand, great number of zero-demand periods and large variation in quantity.[8]

The demand category is determined by the squared coefficient of variation of demand CV2 and Average inter-Demand Interval ADI. See the above mentioned papers for more details.

Features Average demand interval, CV2 and forecastability (demand category) are added.

AGGREGATIONS

We append average demand per week per item and average demand per item. We do not add average demand per month per item, since not all ids were sold over the whole year and for some ids and some months we do not have this information.

Maximum demand per id is added, as well as lower quartile and maximum demand per weekday and per item are added.

Average price per id and a binary variable cheaper than usual are added,

LAG FEATURES

After the training test and test sets are merged Lag features can be calculated. Demand 28, 29 and 35 days ago is calculated and rolling means for lag 28 and windows 7 and 28 are added.

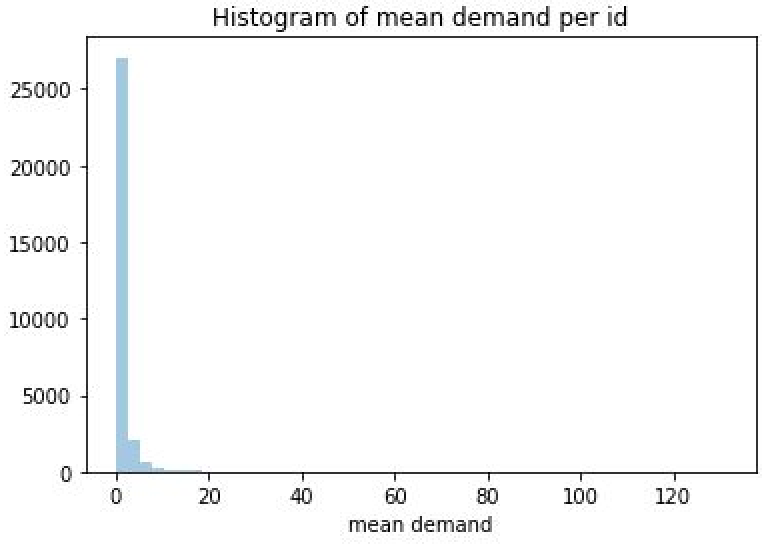
For each id we drop ‘early' rows for which lag cannot be calculated

[7] Syntetos AA, Boylan JE and Croston JD (2005) On the categorization of demandpatterns.J Opl Res Soc56: 495–503.

[8] Costantino, G. Di Gravio, R. Patriarca, and L. Petrella, “Spare parts management for irregular demand items,” Omega (United Kingdom), 2017

TARGET VARIABLE

The target variable ‘demand’ contains 68% of zero which makes prediction a challenging task.



III. IMPLEMENTATION

BENCHMARK

We make naive predictions for the whole data set in order to find out scores for these predictors on Kaggle platform and because there are no parameters to train for the chosen naive predictors.

DEMAND 364 DAYS AGO

For each item the predicted demand is the demand 364 (or 52 weeks) days ago.

The score of this submission is 1.29429

DEMAND ON THE SAME WEEKDAY

For each item the predicted demand is average demand on the same weekday.

The score of this submission is 0.88878

LIGHT GBM

We split the data into train, validation, and test set. The train set contains first 1856 day, the validation set contains 56 days and the test set as defined by competition rules contains 56 days too. Sine the train dataset is quite big (42 644 016 rows and 35 columns) a fast algorithm with low memory requirements is needed. Light GBM is a good candidate for this role.

We train light GBM with the following parameters:

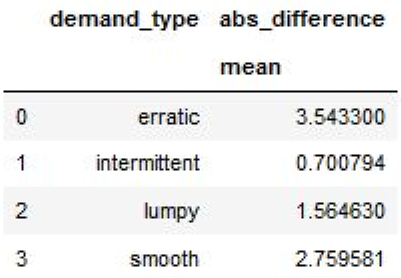
```
params = {      'boosting_type': 'gbdt',
            'metric': 'rmse',
            'objective': 'poisson',
            'n_jobs': -1,
            'seed': 0,
            'learning_rate': 0.1,
            'bagging_fraction': 0.75,
            'bagging_freq': 10,
            'colsample_bytree': 0.75}
```

Since ‘demand’ variable takes non-negative integer values we choose ‘poisson’ objective. RMSSE cannot be chosen as metric for the light GBM, so we stick to RMSE. Other parameters are chosen randomly.

The score of this submission is 0.58584, which is better than the scores of both naive predictions.

IV. RESULTS

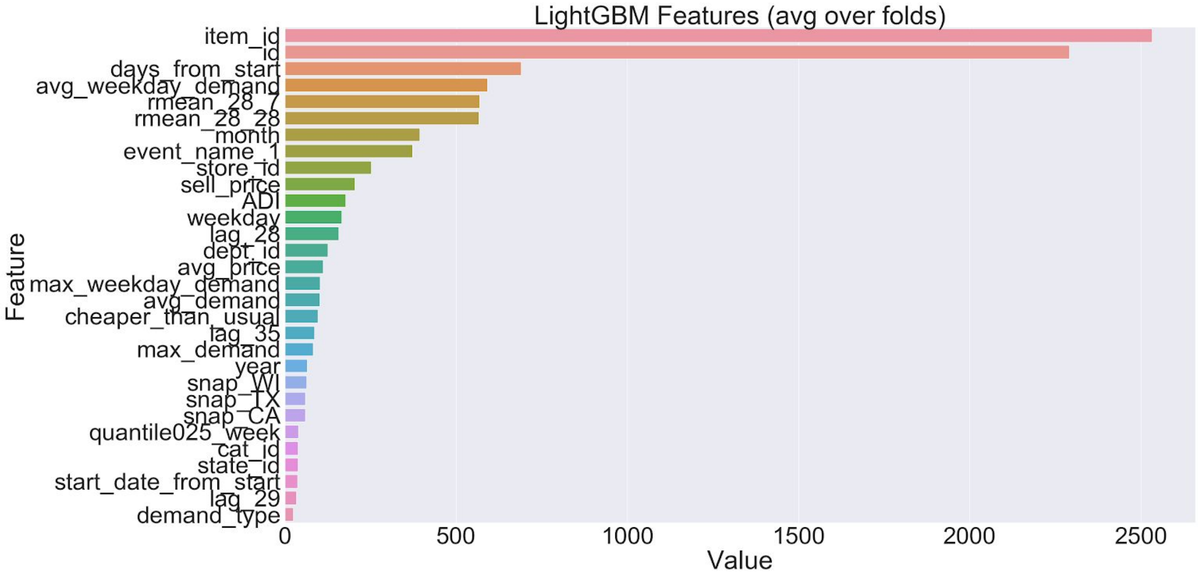
RMSE on the validation set is 2.2247. MAE is 1.05, i.e. roughly speaking we predict one item less or more on average.MAE is distributed over the demand type in the following way



	demand_type	abs_difference
		mean
0	erratic	3.543300
1	intermittent	0.700794
2	lumpy	1.564630
3	smooth	2.759581

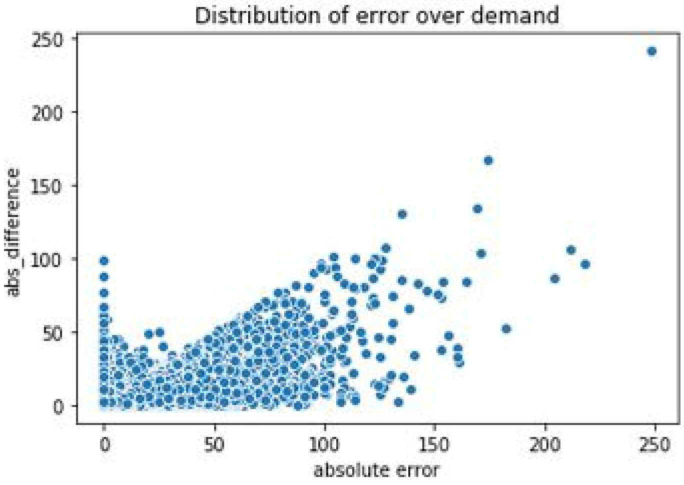
Surprisingly smooth time series has a large mean absolute error. Erratic time series has a large mean absolute error; typically time series of this type is difficult to predict.

Features 'item_id' and 'id' are most important features. Feature 'days_from_start' reflect the trend of data.



V. CONCLUSIONS

Observe the distribution of absolute error over the demand in the validation set.



One can see that error is not uniformly distributed over different demand values, thus there is a room for improvement for the model.

An improvement may be achieved by the following measures:

- Add more features, e.g. std, average month price, other rolling means and lags
- Train separate model for different demand types
- Train separate models for different days, since for the days which are closer to the last day in the validation set, more information available and more recent lags features can be derived
- Tweak parameters of light GBM