Sales Time Series Forecasting
Team A2

Anh Tran (12770698), Sinem Ertem (14616068), Xiaoyu Wu (12538922) and Sebastian Badea (14344203)
Applied Forecasting in Complex Systems 2022
University of Amsterdam

ABSTRACT

This report describes the project in which we used a subset of the M5 Forecasting dataset to build a model that can accurately predict future food sales for the next 28 days at TX3 store in the State of Texas. The data include item level details as well as explanatory variables such as price, promotions, day of the week, and special events. Several models are introduced and their performances are computed by measuring their corresponding RMSE scores. We found that lightGBM is the model with the lowest test average RMSE, concluding its effectiveness in forecasting food sales with this dataset.

1 INTRODUCTION

Having the right amount of products in stock is a challenge for retail because there is a delicate balance between having too much and too little inventory. If a retailer has too much inventory, it can result in unnecessary storage costs and the potential for products to expire or become obsolete. On the other hand, if a retailer doesn't have enough inventory, they may miss out on potential sales. Additionally, having the right amount of inventory can be difficult to predict, as it depends on a variety of factors such as demand, sales trends, and promotions. As a result, accurately forecasting inventory levels is a critical challenge for retailers.

The M5 forecasting challenge could be a solution for the problem of inventory management in retail because it provides a dataset of sales, prices, and promotions data that can be used to develop forecasting models. These models can then be used to predict future demand for products, which can help retailers better understand how much inventory they will need to have on hand.

We will try to forecast the sales of products at one of Walmart's Texas store. Our hypothesis is that during special events or holidays, people tend to have a higher expenditure than usual due to activities such as feasting, special dinners, and reunions. We expect that this increased spending is primarily driven by the celebratory nature of such occasions, which encourages people to treat themselves and their loved ones to a more lavish lifestyle than usual. We further anticipate that this increased expenditure will be more pronounced in certain cultures or demographic groups which place greater importance on such festivities.

This paper presents an analysis of a given dataset for the purpose of forecasting. First, the dataset is explained and an exploratory data analysis is conducted to identify any underlying patterns and trends within the dataset. Then, a range of forecasting methods are used to generate predictions. The results of the models are evaluated and compared to obtain the best possible performance. Finally, the discussion and conclusions are drawn based on the results of the experiment.

2 DATASET

The M5 dataset, provided by Walmart, contains unit sales data for a variety of products sold in the USA. The data is organized as grouped time series, and includes information on 3,094 products that are classified into three categories: Hobbies, Foods and Household. These categories are further disaggregated into seven product departments, and the products are sold in ten stores located in three states: California, Texas, and Wisconsin. For the current analysis, a subset of the data is used, which includes products from the sub-category Food3 sold at the TX3 store in Texas. Overall there are 5 different data sets provided to us which are used:

1. **calendar_afcs2022.csv** which contains information about the dates the products are sold.
2. **sell_prices_afcs2022.csv** which contains information about the price of the products sold per week.
3. **sales_train_validation_afcs2022.csv** which contains the historical daily unit sales data per product starting from 2011-01-29.
4. **sales_test_validation_afcs2022.csv** which contains the historical daily unit sales data per product starting from 2016-04-25.
5. **sample_submission_afcs2022.csv** which contains the number of forecasts to be submitted for point forecasts, exactly 28 days (4 weeks ahead), starting at F1, F2, . . . , F28.

The training data ranges from January 29, 2011 to April 24, 2016. This means that the products in the dataset have a maximum selling history of 1,941 days, or approximately 5.3 years. For the purpose of this analysis, these data were combined into one comprehensive dataset with all variables of interest, by utilizing and adding common columns. The resulting dataset shows - per individual product - the number of items sold per day, along with its price, the day of the week (Monday, Tuesday, etc.), and special events (such as public holidays, celebrations).

item_id	date	sales	...	wday	event_name
FOODS_3_001	2016-04-25	1		3	0
FOODS_3_001	2016-04-26	0		4	0
FOODS_3_001	2016-04-27	0		5	0

Table 1: A Small Overview of the Dataset

3 EXPLORATORY DATA ANALYSIS

Understanding the data available and how variables are associated with one another or with sales is the first step in choosing the variables used in forecasting models. Therefore, to obtain a thorough comprehension of the information and how it relates to our hypothesis, the exploratory data analysis will be carried out from two perspectives: historical time series sales analysis and predictor variable analysis.

Time Series Analysis

In order to gain an understanding of the data, we first select 10 randomly chosen time series from our training sample (see Figure 1).

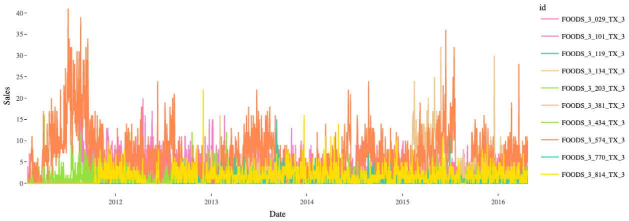

Figure 1: Sales for 10 Randomly Selected Samples

From Figure 1, we observe that most time series feature relatively low daily count statistics together with very small numbers of sporadic spikes. On the one hand, the sales of food appear random, and there is no apparent pattern to the way these individual item-level time series were distributed. Additionally, if we examine each individual item's time series, we can conclude that every item has intermittent unit sales with lots of zeros.

After inspecting some of the individual time series, we can now perform aggregation on all products to obtain general statistics on the data. For example, Figure 2 showcases the total sales per day using *autoplot()*.

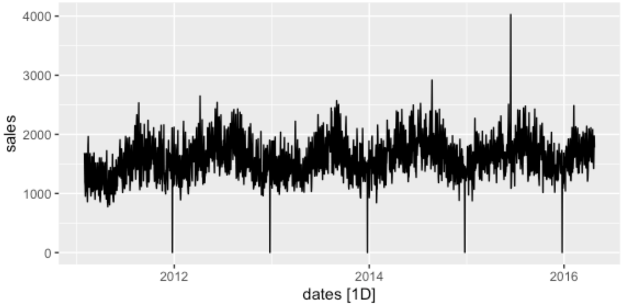

Figure 2: All Aggregate Sales

There are also strong yearly and monthly seasonality as well as a drop at Christmas, the single day of the year when stores are closed. There is also a surprising surge in 2015-06-15 as well. In order to generalize the seasonality and trend, we can remove the Christmas dips and the spike as they could be distracting, and get a smoothing fit curve shown in Figure 3.

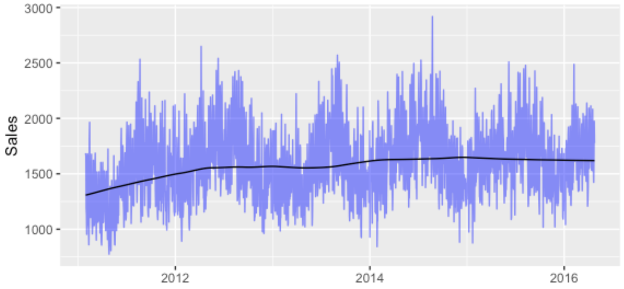

Figure 3: Total Sales with Smoothing Fit

We can observe that the total sales has a sight increase till 2012. After that, the sales becomes consistent, with a constant variance and no clear trend. The ACF also indicates a pronounced seasonality, as the graph is sinusoidal with significant large spikes (seen Figure 4)

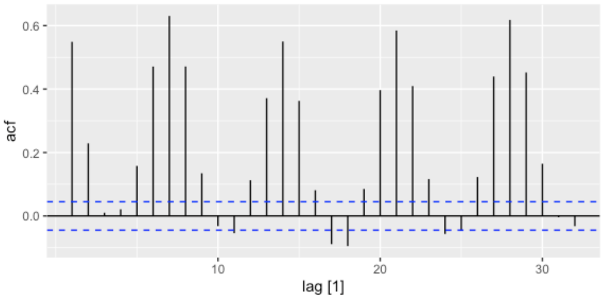

Figure 4: Seasonal Lags in Aggregated Sales

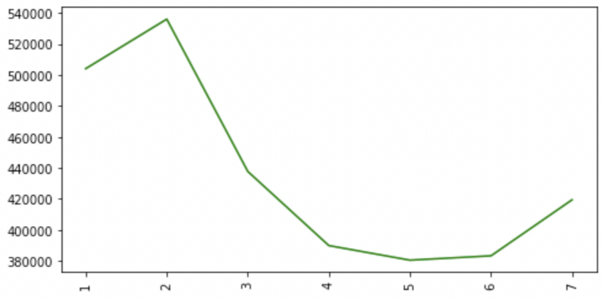

Figure 5: Sales per Weekday Starting from 1 = Saturday

We can investigate smaller scale seasonality further. For example, the line graph in Figure 5 shows that the weekly pattern is strong, with Saturday and Sunday standing out; Monday also appears to gain slightly from the weekend influence.

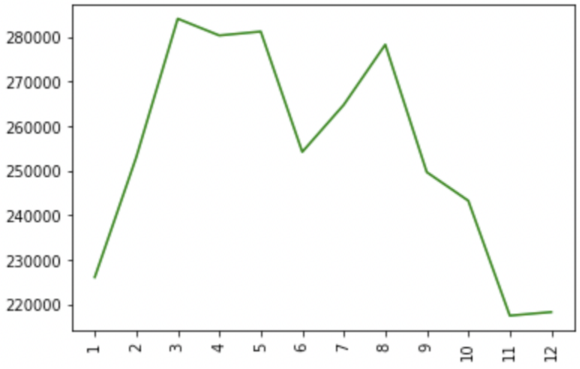

Figure 6: Sales per Month Starting from 1 = January

The monthly pattern in Figure 6 demonstrates that the winter months, especially November and December, exhibit distinct declines, whereas the summer months, with the exception of June, indicate an increase in sales.

Overall, the time series analysis reveal that each item-level time series don't exhibit any significant seasonality or trend, and their sales appear to be randomly distributed-with many zeros and a few spikes. When combining the individual sales, however, strong seasonality is seen on an annual, monthly, and even weekly basis. Therefore, it would be required to take the seasonality into account when predicting individual sales.

Predictor Variable Analysis

In this section, we will focus on the explanatory variables we've been given: item prices and calendar events.

The item price is summarized below. The costs range from 0.02 to 19.48 dollars, with the majority of them falling between 1.88 and 3.5 dollars under that markup. As a result, a product's price may not have an significant impact on its sales because the price is usually affordable and doesn't show a lot variance. Additionally, the individual item pricing are also consistent and don't fluctuate for a very long time-span.

	item price
Min	0.02
1st Quarter	1.88
Median	2.50
Mean	2.85
3rd Quarter	3.50
Max	19.48

Table 2: The statistical summary for food sell prices

On the other hand, there are about 30 events per year in the calendar, with 34% religious events, 32% national events, 32% cultural events and 11% sporting events (see Figure 7).

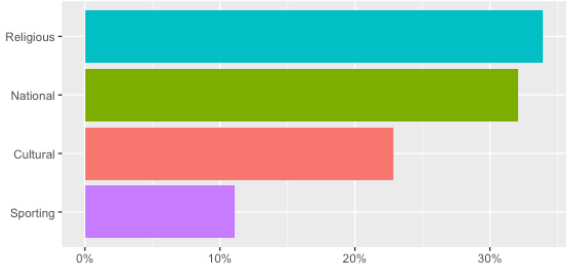

Figure 7: The percentages of various kinds of events

Furthermore, with the exception of the Super Bowl and Christmas, sales are roughly comparable for each event (see Figure 8). The sharp drop during Christmas is because of the fact that all stores are closed. In contrast, the significant increase in sales during the Super Bowl is because Super Bowl is one of the most popular annual sporting events in America which high food consumption.

Additional conclusion can be drawn if we look at days prior and after the event. For example, Figure 9 depicts the relationship between sales and event in February. In general, the days of the events don't always display their peak, but the days prior often do. People are purchasing more in advance of the events for preparation purpose.

The analysis of potential explanatory factors reveals that the impacts of item prices on sales are less important than those of events and holidays because their reasonable and constant prices. The events/holidays, on the contrary, indicates a strong correlation with sales. In particular, regardless

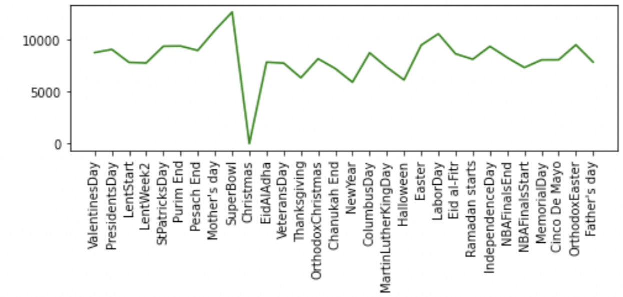

Figure 8: The Sales per Event

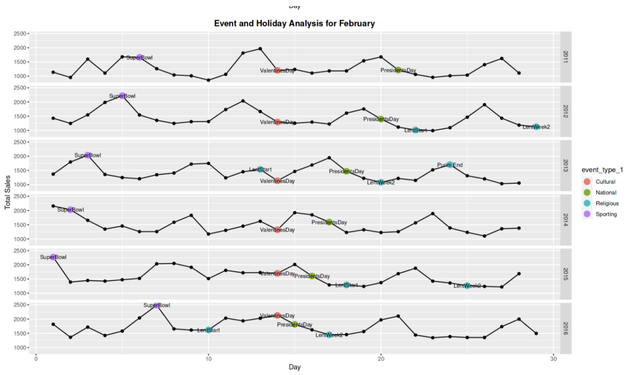

Figure 9: The Sales per Event in Feburary

of the kind of event, the day before the actual event typically results in an increase in sales as people prepare food for the occasion. Sales typically decline a little on the actual event/holiday day since people are spending the day celebrating rather than shopping. Therefore, it would be crucial to take the events' dates into account for accurate forecasting.

4 FORECASTING METHODS

In this section we will first briefly address Feature Engineering which is an important step before using any model and then describe in some detail how and why particular models were selected in the *Model Selection* section.

Feature Engineering

Feature Engineering is a critical step before using any of the forecasting methods. This is the process of transforming existing data into features that better represent the underlying problem to be solved.

(1) Column *event_type_1* and *event_type_2* are converted to binary values. When an event occurs the value 1 is assigned and when no event occurs, the value 0 is assigned. All events are treated equally this way.
(2) Since strong seasonality is seen on both monthly and weekly basis, we take this into account for our prediction. Monthly values are assigned from 1 to 12 and weekly from 1 to 7, with 1 being a Sunday.
(3) In addition to (2) we have added another column called *Weekend* which is a dummy variable depicting whether the day is part of the weekend or not.

Model Selection

Defining an appropriate model for the data is essential for appropriate forecasts. In this section, we will reason the models we chose by examining its complexity and its compatibility with the previous explanatory data analysis.

We will first start with benchmark models which were also introduced in the lectures such as *Mean/Naive/Seasonal Naive/Drift*. Next, we will explain the more complex models such as *ETS and Dynamical Regression*.

Mean. While the mean is a really simple approach it is an effective baseline model. The prediction of all future values is calculated by taking the mean of the values recorded in the specific time series. However, we have to take into account that this method also has its limitations because we have seen during the EDA that a lot of products were not sold anymore after a certain time, or even products being introduced later on. this means that we do not have a completely honest view of the sales of some products.

Seasonal Naive. Seasonal Naive is another benchmark model we have used for forecasting. If the data shows seasonality related to the day of the week, the seasonal naive method can be considered useful for forecasting this type of data. As we have seen in the EDA, days like Saturday and Sunday show a strong peak.

Naive. Simple forecasting techniques like the naive method, which can be unexpectedly effective, especially in many economic and financial time series. The Naive method gives all weights to the latest observation, so the forecast of all future values is the final value of observation.

Drift. A different approach to the naive method is to let the forecasts increase or decrease as time passes. This amount of change with time (known as the drift) is set to be the same as the average change seen in the past data.

Dynamic Regression. Dynamic regression integrates an ARIMA model which looks at information from past observations and a regression model which looks at information from predictors. We want to be able to capture time series dynamics, while also using events as a predictor, therefore a dynamic regression model is appropriate.

ETS. Exponential smoothing (ETS) is a widely used time series forecasting method. Rather than giving all weights to the latest observation, exponential smoothing weighs later observations more heavily but not totally disregarding previous ones. It is useful for forecasting data with no clear trend which is suitable for this task as the individual item-level time series doesn't exhibit trend.

LightGBM. LightGBM is a gradient boosting framework that uses a tree-based learning algorithm. It is designed as a fast method for forecasting while providing great accuracy. It works by training weak decision tree models, then combining them to form a stronger model through boosting.

Performance Metrics

When selecting models, it is standard practice to segment the available data into training and test data. The training data is employed to calculate any parameters of a forecasting method, while the test data is utilised to assess its accuracy. As the test data is not used to anticipate forecasts, it should render a reliable evaluation of how effective the model is likely to be when forecasting new data.

The two most widely employed scale-dependent metrics are based on absolute and squared error metrics[1]:

$$MAE = \sum_{i=1}^{D} |x_i - y_i| \tag{1}$$

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(\frac{d_i - f_i}{\sigma_i}\right)^2} \tag{2}$$

As per requirements, the RMSE of our models obtained from testing on the **test validation dataset** will be used to identify the best model. This metric is also appropriate as all models are estimated on the same unadjusted time series. To make sure that we do not have a biased test set, for each model, RMSE was calculated as the average RMSE of point forecasts per product.

5 RESULTS

In previous sections we have introduced several models from simple to more complex. In this section the results of all the used models are discussed and compared using RMSE as explained in the *Performance Metrics* section.

Benchmark models

While discussing the results, it is important to distinguish between models trained and tested on the dataset without Feature Engineering and with Feature Engineering. Since *with* Feature Engineering we included variables such as event to see if this can give a valuable prediction.

Table 3 shows all of the benchmark results according to the RMSE metrics measured on the test set *without* Feature Engineering (left Table) and *with* Feature Engineering (right Table).

Model	RMSE
Mean	3.421
Seasonal Naive	3.680
Naive	4.108
Drift	4.135

Model	RMSE
Mean	2.056
Seasonal Naive	2.322
Naive	2.340
Drift	2.351

Table 3: Results benchmark methods without FE (LEFT) and with FE (RIGHT)

Figure 10 shows an example of how the four benchmark methods are performing on a randomly chosen product, together with the 80% and 95% prediction intervals. Figure 11 however shows an example of the four improved benchmark methods and how they are performing on another randomly chosen product.

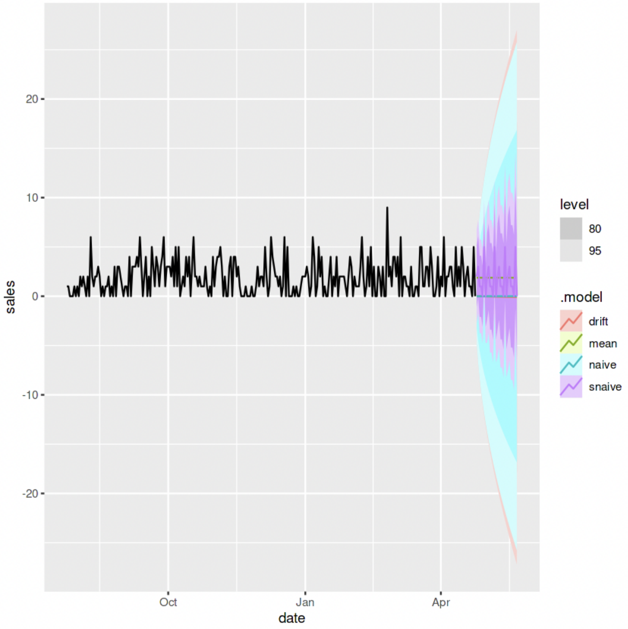

Figure 10: Example forecast: benchmark methods

The results demonstrate in the first place that the application of Feature Engineering is effective in decreasing the RMSE for the benchmark models. In section 4 (*Feature Engineering*) we listed all of the methods used for Feature Engineering. Therefore we can conclude that including variables such as *Events* have positive influence on the RMSE for the benchmark models. For this reason we have chosen to continue testing out other (more complex) models with application of Feature Engineering.

Another interesting finding is that the *Mean* model performed better than all of the other benchmark models. However, this is not really surprising as the Mean model averages over past values to predict the future while e.g. Seasonal

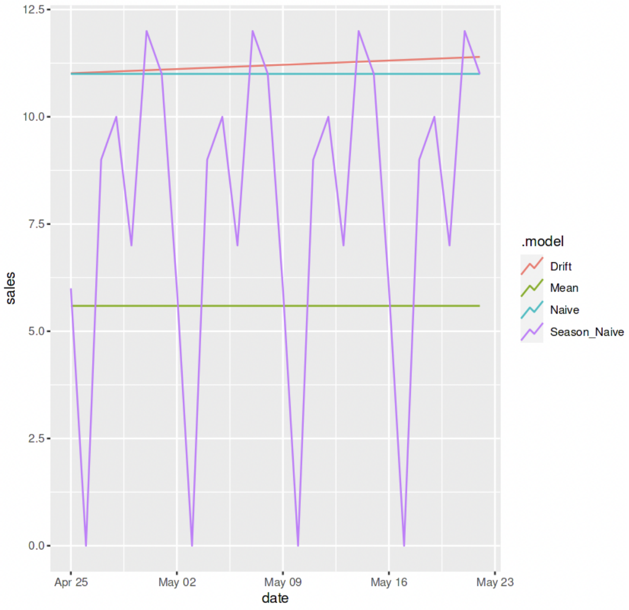

Figure 11: Example forecast: benchmark methods

Naive is being applied on our dataset which has inconsistent seasonality patterns. Therefore we define our baseline model as the Mean and the rest of the models are being compared against the RMSE of the Mean which is: *2.056*.

Complex models

The benchmark model with feature engineering is now compared to other complex models:

Model	RMSE
Mean	2.056
Dynamic regression	1.829
lightGBM	1.763
lightGBM (rounded)	1.804
ETS	1.893

Table 4: Results of different complex models

As can be seen from the table, lightGBM is the model with the lowest test average RMSE. We therefore conclude that this is the most appropriate model for forecasting the time series at hand. To further test the model, we've also calculated the RMSE of the rounded predictions, as this problem only deals with integers. In this scenario we also highlight the additional benefit of lightGBM being a fast algorithm, making it easier to tune the hyper-parameters for the task.

6 DISCUSSION

According to the point forecasts given by our best performing model, the Walmart store TX3 can expect to sell approximately 48823 items in the Food3 category from 25-04-2016 to 22-05-2016. If we choose to round the numbers, then the total becomes 47918 instead. While the lightGBM model has good accuracy, this method is not without limitations.

One limitation from our methods is that all of them predict continuous values. Because of that, in order to give realistic forecasting predictions, we need to process them by rounding the numbers to the nearest integers.

Other limitations have to do with the variables included in the analysis. More explanatory variables would have been useful for effectively forecasting, such as product kind. Some products may respond well to holidays, while others may be heavily influenced by price changes. Therefore, a generalized model attempting to forecast every product with a single standard is insufficient. Future analysis could group products by type and find a suitable model for each type to improve forecast accuracy.

In a similar vein, the data subset used in this analysis does not include different store locations. Location could be an important predictor because each product might perform differently depending on where they are sold - some might be more in-demand in a given region than others. Thus, taking into account store locations would produce more realistic forecasts that is based on more comprehensive real-life demand, leaving less chance for a store to keep stock of an unpopular product.

7 CONCLUSIONS

Our hypothesis is that during special events or holidays, people tend to have a higher expenditure than usual and the aim of this research was to find out whether those events could help us forecasting.

Through an Exploratory Data Analysis (EDA), a strong correlation between events/holidays and sales was observed. Specifically, it was noted that sales increased the day before the event/holiday, likely due to people acquiring necessary items for the upcoming celebration. Conversely, on the actual day of the event/holiday, sales decreased as people spend their time celebrating instead of shopping. As such, it was essential to consider the dates of events/holidays when forecasting sales.

We started training and testing with several benchmark models such as Mean, Seasonal Naive, Naive and Drift. Initially, we trained disregarding the events columns, and the Mean model yielded the highest score (RMSE: 3.421). By training the benchmark models with the events columns, we observed a decrease in the RMSE score (RMSE for Mean model: 2.056). This suggests that events have a beneficial effect on forecasting, and thus we utilized the events columns when training the more complex models. Another interesting finding is that the Mean model outperformed all other benchmark models, which highlights its ability to generate useful predictions by averaging over past values.

For the complex models we have implemented Dynamic Regression, lightGBM and ETS. LightGBM has the lowest RMSE (1.763), making it the most suitable model for this task. Furthermore, lightGBM is a fast algorithm, which made it easier to optimize the hyper-parameters for the task.

REFERENCES

[1] Hyndman, R.J., Athanasopoulos, G, "Forecasting: principles and practice, 3rd edition, OTexts: Melbourne, Australia."; 2021 OTexts.com/fpp3.

[2] Microsoft Corporation, "Official lightGBM documentation"; 2022 https://lightgbm.readthedocs.io/en/v3.3.2/

[3] DataTechNotes, "LightGBM Regression Example in R"; 2016 https://www.datatechnotes.com/2022/04/lightgbm-regression-example-in-r.html