M5 Forecasting based on LightGBM

LU Fei 20797386, XIONG Yi 20787642, ZHENG Hao 20797594

Abstract

If we can know the sales volume of goods in the next period, stores will be able to make greater profits. Therefore, many stores are looking for a way to accurately predict sales. In the M5 forecasting project, we need to use nearly 2000 days of sales data to predict the sales in the next 28 days. We first preprocess the data and observe the characteristics of the data through visualization. Then we utilize the LightGBM model and adjust the parameters to the best state to predict the next 28 days' sales. Our final prediction obtains a private score of 0.62812.

1 Introduction

1.1 Problem Definition

Time sequence prediction is a very popular topic in both scientific and real world. The accuracy of prediction usually relies on the historical data and current features. In this project, we use detailed data from Walmart, a chain store, trying to obtain a relevantly great prediction accuracy of the next 28 days' sales. We take some important features into our model consideration such as the weekly/monthly trend of the sales and if the dates are the ones of some events. As for the model selection, we use LightGBM to train and to predict.

1.2 Dataset Description

The dataset is provided by Walmart. It covers sales of stores in three US States, i.e. California, Texas, and Wisconsin and 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. Also, it provides the calendar data to make it possible for us to join the sell prices table and sales_train_evaluation tables together with the attribute "wm_yr_wk" and "d_i". Here's the outline of the data.The figures are shown in figure1. The sales_train_evaluation contains 30490 unique item values and for each item, we can get the sales data from d_1 to d_1941, which is from 2011-01-29 to 2016-05-22. The goal is to predict the following 28 days' sales.

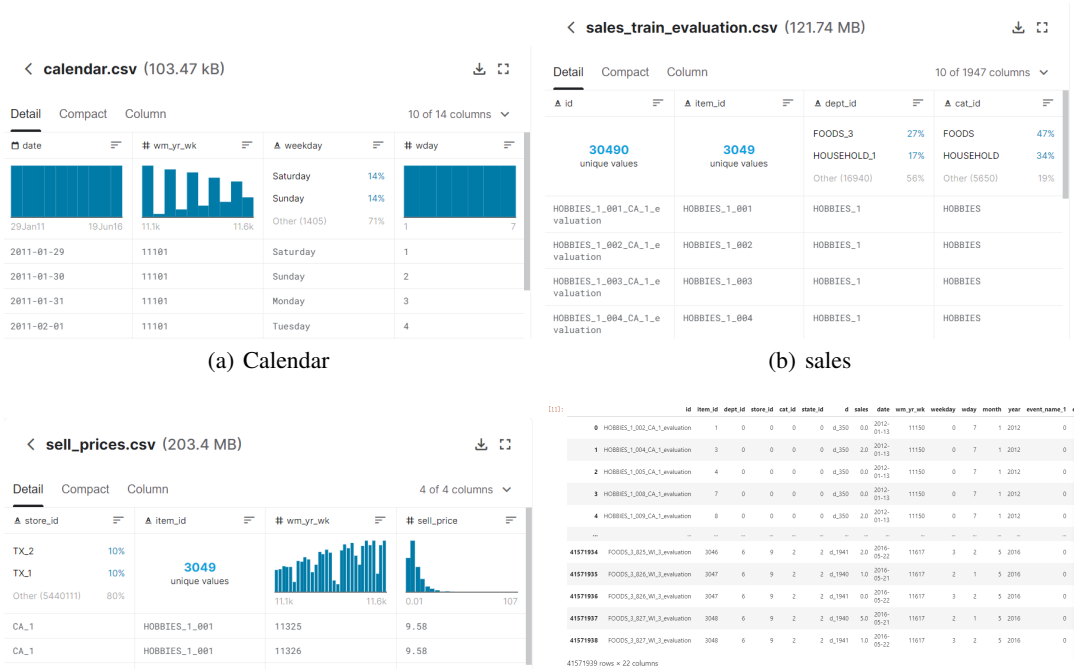
2 Exploratory Data Analysis

The data used this project are sales data from the real world. In order to select an appropriate algorithm to do the prediction, we must study the data first. Therefore, we do exploratory data analysis [1] to find out the relation between each feature and the sales.

2.1 Data Preprocessing

To better extract the features, we need to join tables together using the common attributes. In sales file, each row contains sales from d_1 to d_1941, which is hard to analyze. We use pandas.melt() to convert those column to row values. Then we use pandas.merge() to connect with calendar table and the sell price tables based on some common attributes. The final table shapes like (d) in figure 1.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.



(a) Calendar (b) sales

(c) prices (d) final

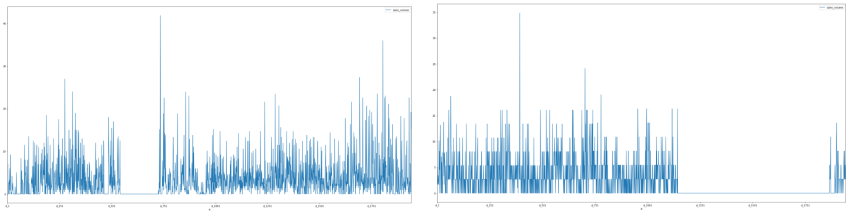
Figure 1: Dataset outline

2.2 Sales trend

2.2.1 Sales trend with specific items

First, we visualize the sales of a specific item in a specific store over time. Through visualization, we find that there are a large number of 0 values in item sales, and there is no obvious law. Some items didn't sell at first, but then they began to sell at sometime. Some items were sold at first and then stopped selling. Some items have not been sold for some time. As shown in the figure 2, (a) is figure of id HOBBIES_1_008_CA_1_evaluation, we can find that there was no sales of for a period of time, and we can infer that it has stopped selling for this period of time.

The situation is the same for an item sales in the whole market. A large number of 0 sales, i.e. stopping selling, may occur in any time period, and the sales of each item is not regular. As shown in the figure 2, (b) is the figure of item_id 107.



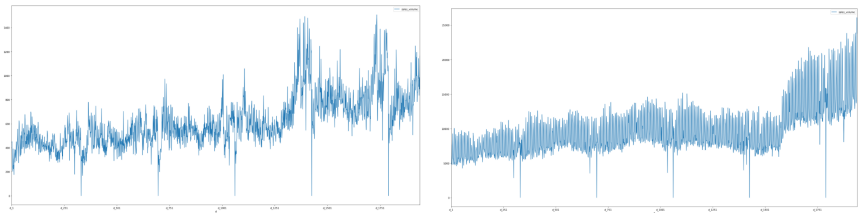
(a) id: HOBBIES_1_008_CA_1_evaluation (b) item_id: 107

Figure 2: Sales of id and item_id

2.2.2 Sales trend with specific departments or stores

Next, we visualize the sales volume of departments with time. We find that the sales volume of each item department has a certain cyclical law, and gradually increases with time. As shown in the figure 3, (a) is dept_id 1.

It shows a same pattern for stores. We find that the sales volume of each stores has a certain cyclical law, and gradually increases with time. As shown in the figure 3, (b) is store_id 1.



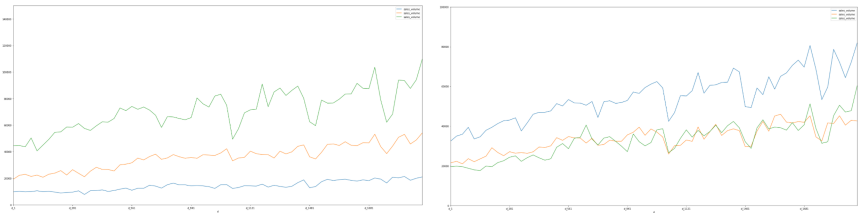
(a) dept_id: 1 (b) store_id: 1

Figure 3: Sales of dept_id and store_id

2.2.3 Sales trend with product categories or states

Figure 4 (a) is an example of the sales volume of all categories (3 in total). Since there are multiple function lines, function points are drawn every 4 weeks (28 days) to make the image clearer. Among them, green is food, orange is household and blue is hobbies. It can be seen that the sales volume of food is much higher than that of the other two categories, and household has more than hobbies. The sales of the three categories have a certain time law and increase with time.

Figure 4 (b) is an example of the sales volume of all states (3 kinds in total). Since there are multiple function lines, function points are drawn every 4 weeks (28 days) to make the image clearer. Blue is CA, orange is TX and green is WI. It can be seen that the sales volume of Ca is much higher than that of TX and WI. TX is roughly the same as WI. At first, the sales volume of TX is a little more than that of WI. Finally, the sales volume of WI exceeds that of TX. The sales volume of the three states has a certain time periodicity and increases gradually.



(a) FOOD, HOUSEHOLD and HOBBIES (b) CA, TX and WI

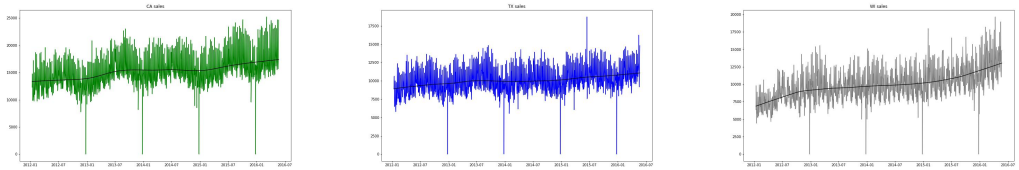
Figure 4: Sales of cat_id and state_id

2.3 Weekly days & Monthly days

In the beginning, we draw the relationship figure between time and the total sales and present a trend curve. The general curve is quite periodic, with slow increment and the sales reached to the bottom at almost every end of the year. The increment may be because of the economy and the bottom point maybe caused by that at the end of the year, stores may shut down or shorten the opening time. We then draw the sales curve of each state respectively. The general trend is similar, but the state CA has more sales than other two states, the state WI has the fastest increment above all and in state TX, it has a relatively peak point in 2015-07 while others don't have such an extreme peak. Apparently, the sales go with the time, so we will divide the time dimension in two respects, weekly and monthly.It is shown in figure 5.

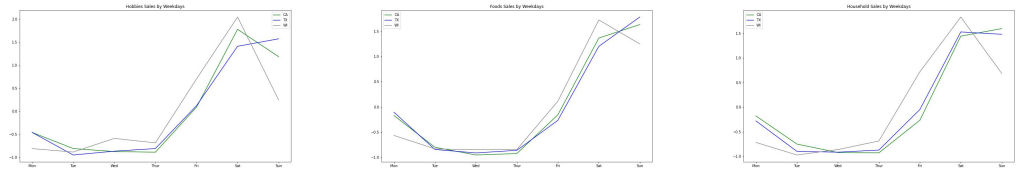
2.3.1 Weekly trend

We then try to figure out the weekly time's effect on the total sales at first. We divide the category and try to find out if different category items will have different sales trend. There are three curves in each figure, representing each state, i.e. CA, TX, and WI. The general trend seems make sense, with the peaks almost at the weekends and in the middle of the weekdays, the sales remains low. However, the state WI's sales decrease sharply on Sundays while other states remain increasing.It is shown in 6.



(a) CA (b) TX (c) WI

Figure 5: Sales trend



(a) Hobbies (b) Foods (c) Household

Figure 6: Weekly trend

2.3.2 Monthly trend

When it comes to the monthly days, the sales curves also have similar trends, but they have some outstanding features as well. People seems to like shopping in spring and the sales will fall down when May is approaching. The curves of hobbies are quite different from those other curves between June and October, presenting a first down then up trend while others are totally opposite.7

(a) Hobbies (b) Foods (c) Household

Figure 7: Monthly trend

Finally, we draw the heatmap to find the co-effects of weekly features and monthly features toward total sales data. The figure 8 shows that sales are lower when it is winter or in the middle of the year.

From the analysis above, we can tell that the sales are relevant to the weekly and monthly features. So we may use these features as input to train the model and predict the results.

2.4 Event days & SNAP days

2.4.1 Sales in event days

We assume the sales during the event days will be different from the normal days, and then we do some exploratory data analysis to find out the relation. The following two figures9 show how the 3 categories of items (FOODS, HOUSEHOLD, HOBBIES) sold and how the sales in 3 states (CA, TX, WI) in event days compared with in normal days. As shown, in the event days, the sales of the 3 categories of items and the sales in the 3 states were all lower than in the normal days, maybe it's due to the stores' closing during the event days.

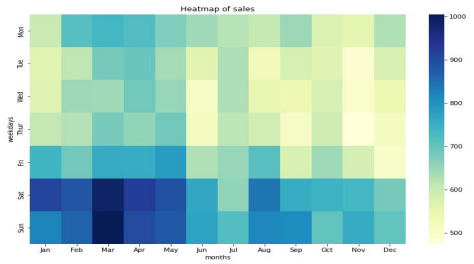


Figure 8: heatmap

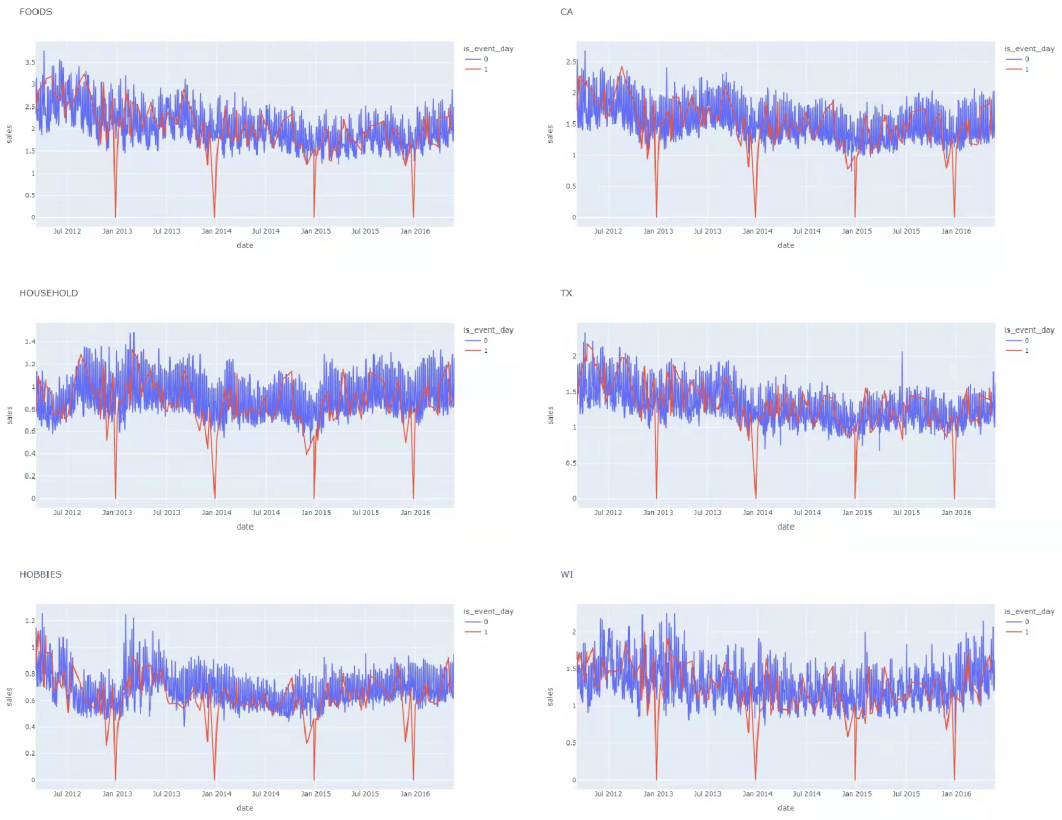


Figure 9: sales in event days and normal days

2.4.2 Sales in SNAP days

We assume the sales during the SNAP days will be different from the normal days, since the coupon can be used during these days. The following two figures10 show how the 3 categories of items (FOODS, HOUSEHOLD, HOBBIES) sold and how the sales in 3 states (CA, TX, WI) in SNAP days compared with in no-SNAP days. As shown, in the SNAP days, the sales of the 3 categories of items and the sales in the 3 states were all higher than in the no-SNAP days.

2.5 Conclusion

After the exploratory data analysis, we find some relation between the sales and some data attributes. We pick out the date features, regional features, event/SNAP features to train our model.

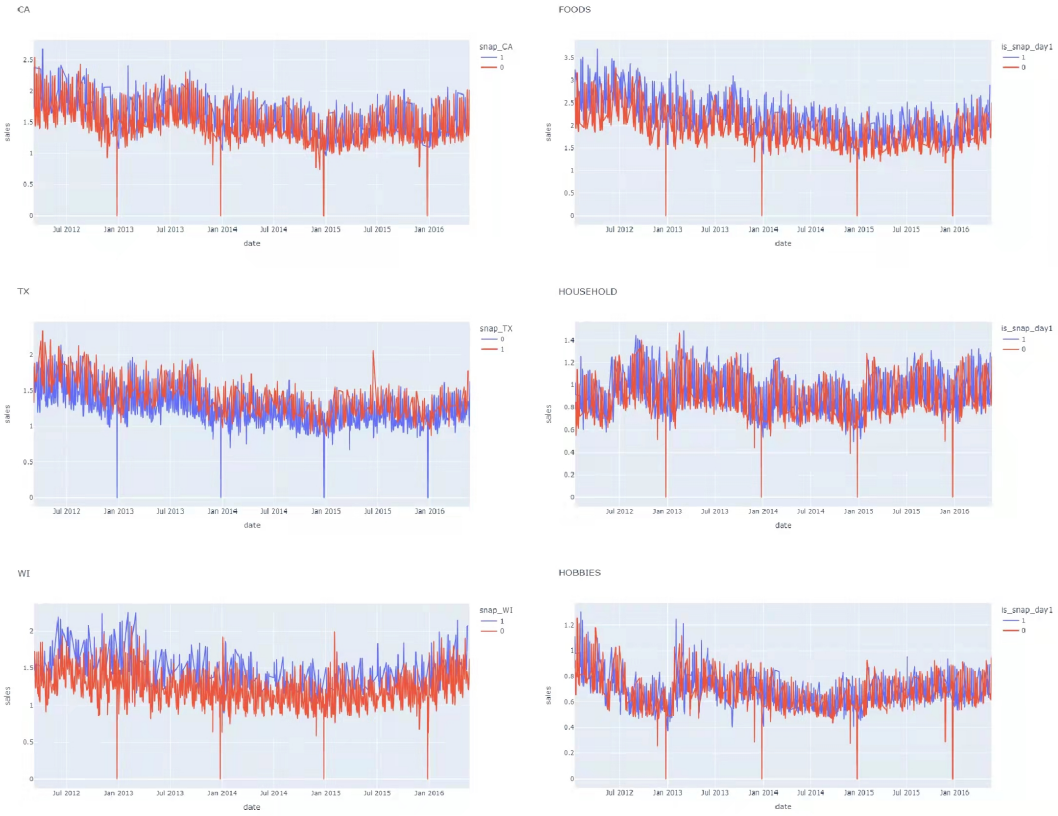


Figure 10: sales in SNAP days and no-SNAP days

3 LightGBM and Model Train

Since the goal of this project is to predict the future sales and there are many features showing impact to the sales, we choose LightGBM, a powerful regression model, to train this large amount of data and do the prediction.

3.1 Basic Introduction of LightGBM

LightGBM[3] is an improved version of GBDT. LightGBM improves the shortcomings of GBDT. It is a distributed and efficient model, which can be trained faster.

3.1.1 Gradient Boosting Decision Tree (GBDT)

GBDT consists of three concepts: Regression decision tree (DT), gradient boosting (GB), and shrink (an important evolution). When branching, the Regression decision tree will exhaust each threshold of each feature to find the best segmentation point, and the measurement standard is to minimize the mean square error. The core of GBDT is to accumulate the results of all trees as the final result. Each tree of GBDT updates the target value with the residual obtained from the previous tree, so that the sum of the values of each tree is the predicted value of GBDT. Shrinkage sets a weight for each tree, which is multiplied when accumulating. When the weight decreases, the number of base models will increase. In this way, the results can be approached in small steps to avoid over fitting.

The pseudo-code of GBDT is shown in figure 11. Specifically, the algorithm steps are as follows:

1. Initialization, estimate the constant value that minimizes the loss function. It is a tree with only one root node, that is, ganma is a constant value.

2. Calculate the value of the negative gradient of the loss function in the current model and take it as the estimation of the residual; Then, the node region of the regression leaf is estimated to fit the approximate value of the residual; Then, the value of leaf node region is estimated by linear search to minimize the loss function; Finally, update the regression tree.

3. Get the final output model $f(x)$.

Algorithm 10.3 *Gradient Tree Boosting Algorithm.*

1. Initialize $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$.
2. For $m = 1$ to M:
   (a) For $i = 1, 2, \ldots, N$ compute
   $$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}.$$
   (b) Fit a regression tree to the targets r_{im} giving terminal regions R_{jm}, $j = 1, 2, \ldots, J_m$.
   (c) For $j = 1, 2, \ldots, J_m$ compute
   $$\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}(x_i) + \gamma\right).$$
   (d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.
3. Output $\hat{f}(x) = f_M(x)$.

Figure 11: pseudo-code of GBDT

3.1.2 Gradient-based One-Side Sampling (GOSS)

Although GBDT has no data weight, each data instance has different gradients. As we know, instances with large gradients have a greater impact on information gain. So during down sampling, LightGBM use GOSS to try to retain samples with large gradients, and randomly remove samples with small gradients, which make LightGBM obtain more accurate results.

3.1.3 Exclusive Feature Bundling (EFB)

From the perspective of feature reduction, LightGBM uses EFB to bind mutually exclusive features, that is, they rarely take non-zero values at the same time. Usually in practical applications, although there are many features, but because the feature space is very sparse. Especially in sparse feature space, many features are almost mutually exclusive, so LightGBM choose to bundle mutually exclusive features. Finally, the bundling problem is reduced to the graph coloring problem, and the approximate solution is obtained by greedy algorithm.The algorithm steps of EBF are as follows: sort the features according to the number of non-zero values; Calculate the conflict ratio between different features; Traverse each feature and try to merge features to minimize the conflict ratio.

3.2 Model Setting and Training

In the actual model training [2] and testing, in order to predict the sales data from d_1942 to d_1969, we use data from d_1 to d_1941, which are in file sales_train_evaluation.csv, and put them into the model for training.

We have determined the parameter setting of the model through many attempts. Specifically, the learning rate is 0.075, the number of leaves is 128, the number of iterations is 1500, the min data in a leaf is 100, bagging_freq is 1, verbose_eval is 20.

4 Results and submission

The evaluation is based on the Weighted Root Mean Squared Scaled Error (RMSSE) between the true value and prediction value. We used the "sales_train_evaluation.csv" data which includes the

sales from d_1 to d_1941 to train our LGBM model, and predicted the sales on d_1942 - d_1969. According to the scoring system of Kaggle, our score is 0.62812, and ranked nearly 205th of more than 5500 groups on the private leader board. Our team name is "msbd5013_LU_ZHENG_XIONG".

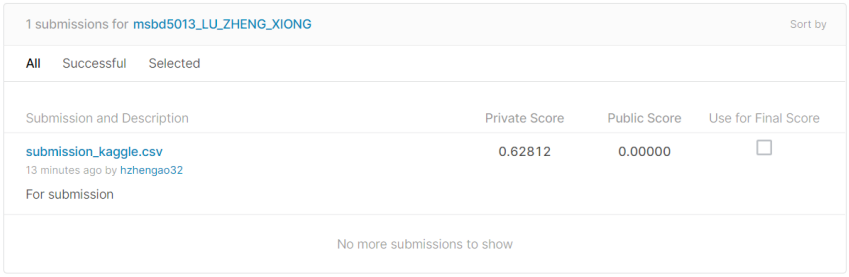


Figure 12: score

5 Further improvements

Applying more features to train our model. In our model, we utilized all of the attributes provided by the original data set. However, there are some crossing features may also influence the sales. In future work, we will try different combination of the current attributes to train our light-GBM model.

6 Contributions

Name	Contribution Descript
LU Fei	Exploratory Data Analysis; Model training; Report writing
ZHENG Hao	Problem Analysis; Exploratory Data Analysis; Report writing
XIONG Yi	Data analysis and visualization; Model training and prediction; Report writing

References

[1] *Exploratory data analysis*. URL: `https://zhuanlan.zhihu.com/p/298834571`.

[2] *Kaggle Notebook*. URL: `https://www.kaggle.com/akashsuper2000/m5-lightgbm-model`.

[3] Guolin Ke et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: `https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf`.