

# 2. Slides.pdf

*by HU Zhenzhuo*

---

**Submission date:** 31-May-2020 05:14PM (UTC+0800)

**Submission ID:** 1335071803

**File name:** 2.\_Slides.pdf (2.33M)

**Word count:** 567

**Character count:** 2813



# M5 Forecasting with LightGBM

Kaggle  
(Walmart)

BY

HU Zhenzhuo  
20665208  
WU Siliang  
20672110  
ZHANG Zhiyi  
20657263

# CONTENS

Background

The Analysis

Performance

Improvements

# 01

## Background

- 
- Description
  - The Dataset
  - Evaluation

# Description



- The objective of the M5 forecasting competition is to advance the theory and practice of forecasting by identifying the way that provide the most accurate point forecasts for each of the 42,840 time series of the competition.<sup>2</sup>
- Extract the feature, randomized the training set, use boost, and finally applied the model to the test set to get the corresponding sales and submit the final result.

# The M5 Dataset

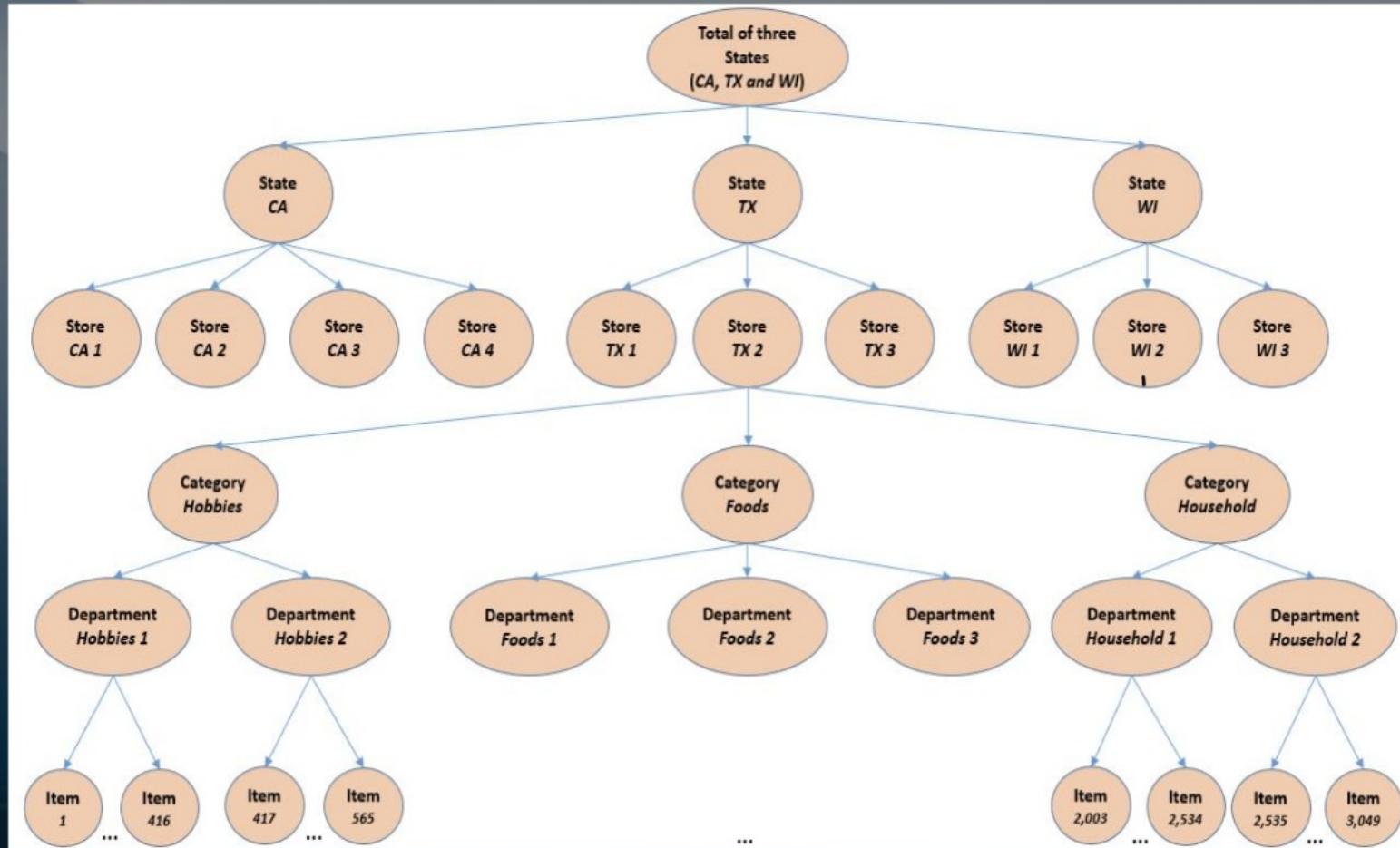


Figure 1: An overview of how the M5 series are organized



## Evaluation

The accuracy of the point forecasts will be evaluated using the Root Mean Squared Scaled Error (RMSE)..

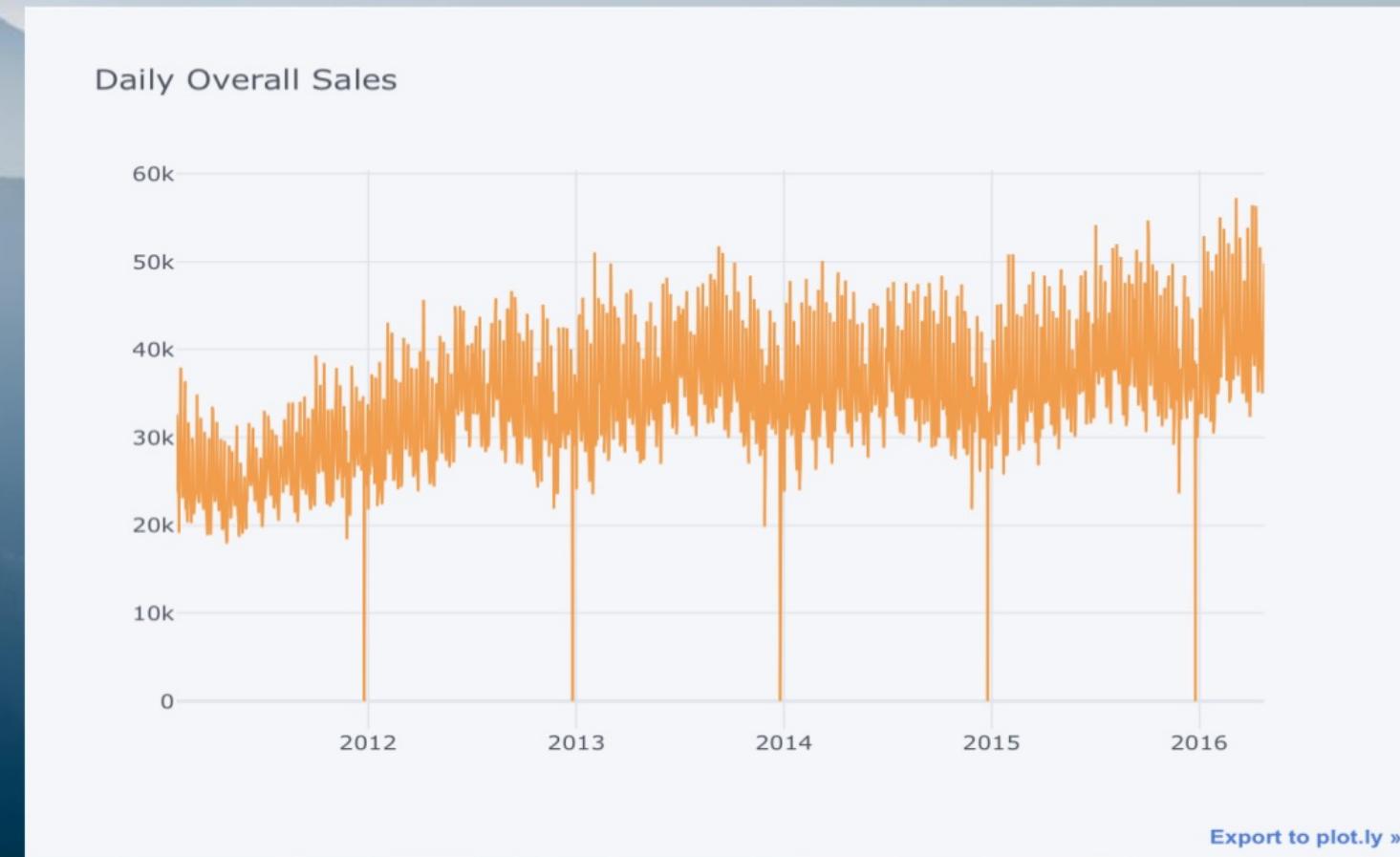
$$\text{RMSE} = \sqrt{\frac{1}{h} \frac{\sum_{t=n+1}^{n+h} (Y_t - \hat{Y}_t)^2}{\frac{1}{n-1} \sum_{t=2}^n (Y_t - Y_{t-1})^2}},$$

# 02

## The Analysis

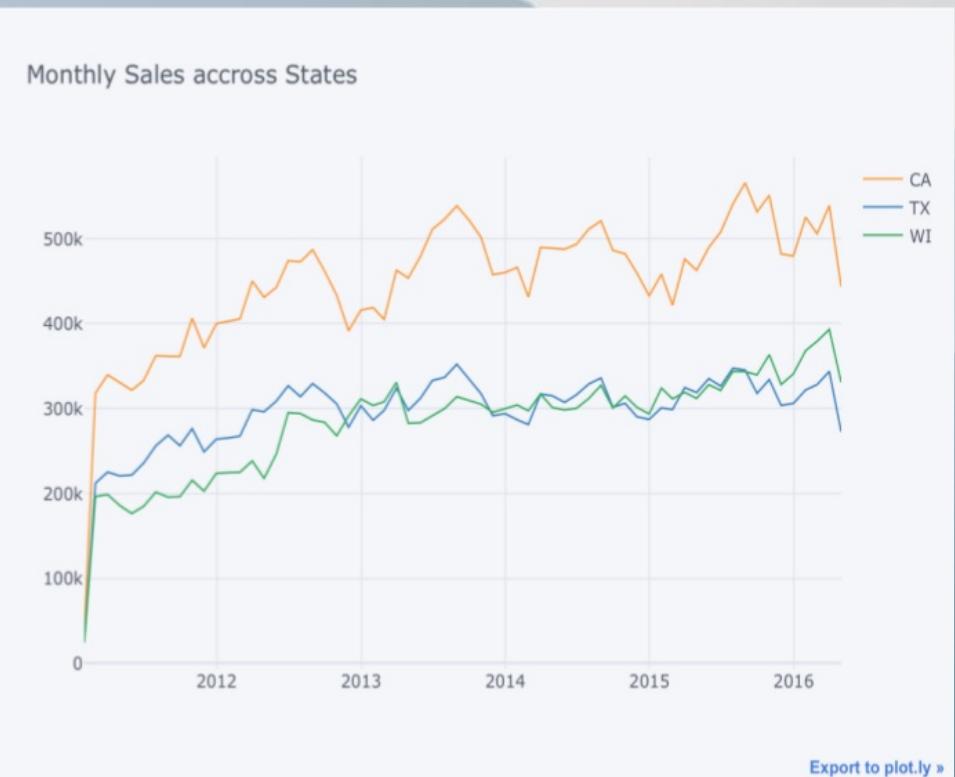
- Exploratory Data Analysis (EDA)
  - Time Series Views
  - Impact of Events and SNAP Days on Sales
  - Analysis on Prices Change
- Feature Engineering (FE)
- LightGBM Model
- Forecast

# EDA: Time Series Views



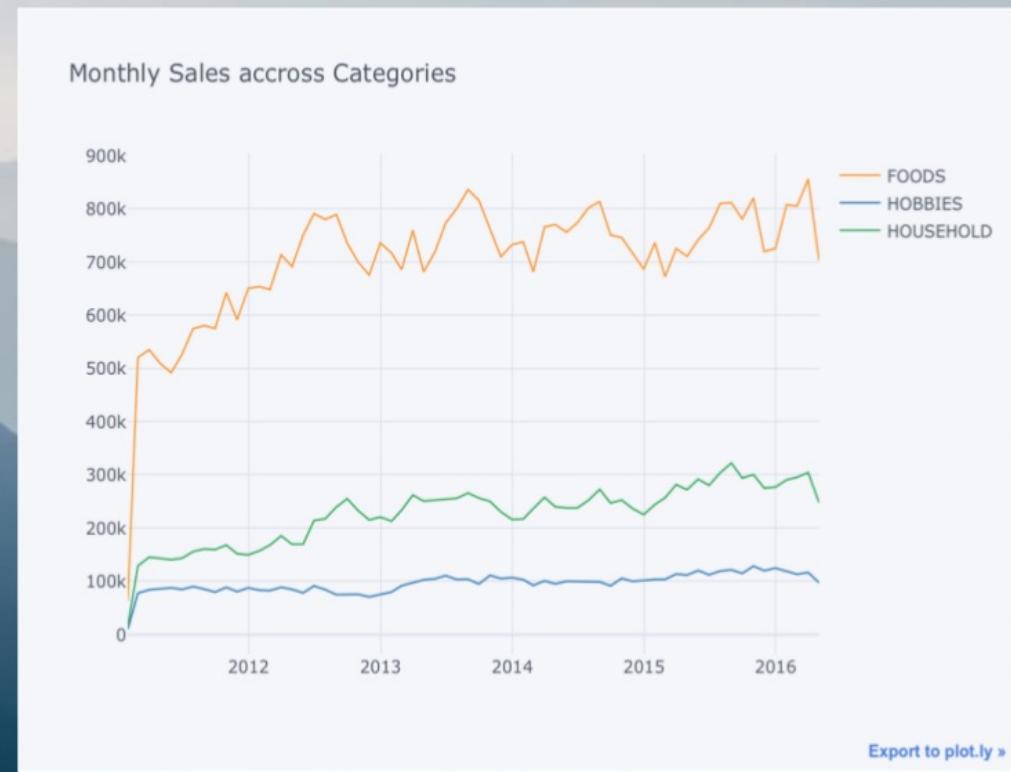
**Figure 2: Daily overall sales**

# EDA: Time Series Views



1

**Figure 3: Monthly sales across states**



**Figure 4: Monthly sales across states**

# EDA: Time Series Views

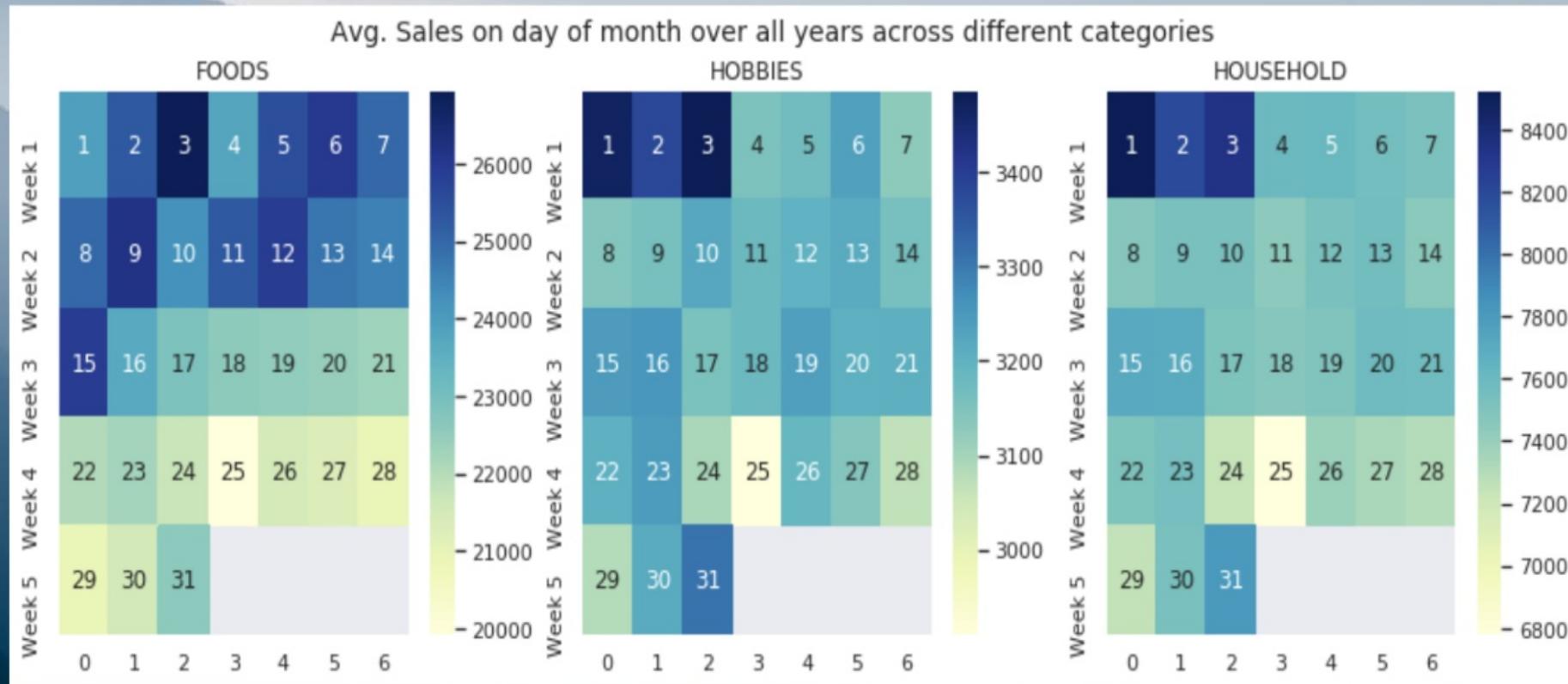


Figure 5: Average sales on day of month over all years across different categories

# EDA: Impact of Events and SNAP Days on Sales

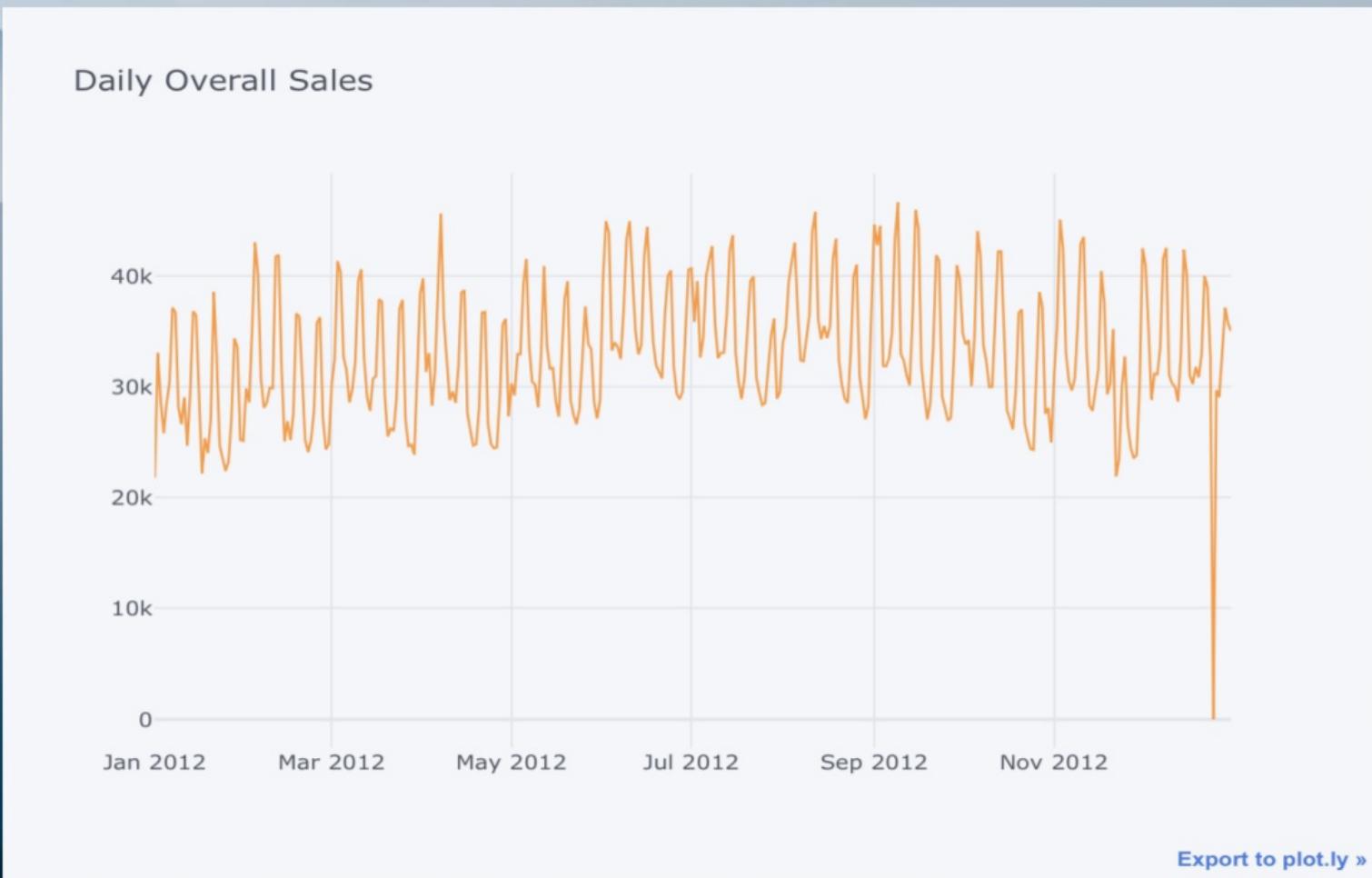


Figure 6: Daily overall sales in 2012

# EDA: Impact of Events and SNAP Days on Sales

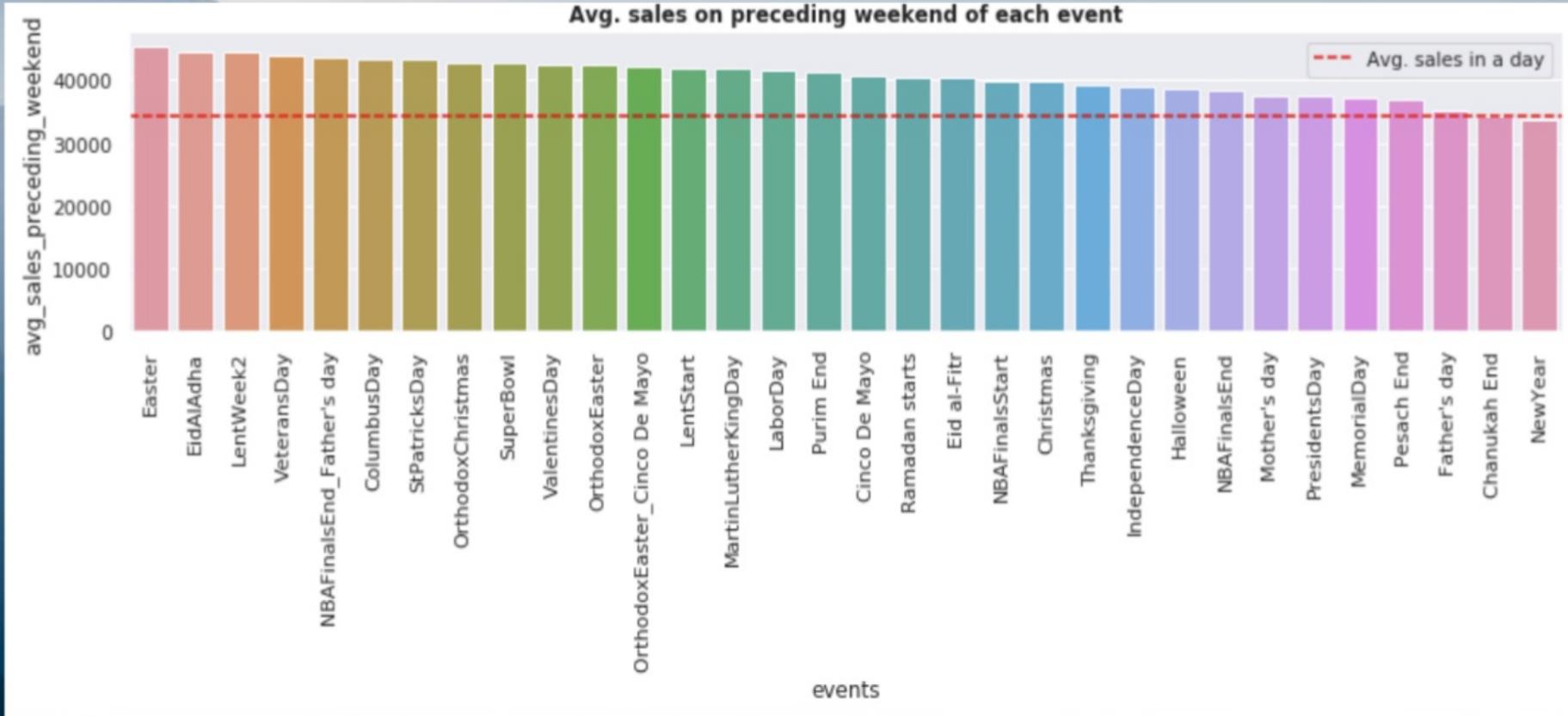


Figure 7: Sales of weekends preceding each event

# EDA: Impact of Events and SNAP Days on Sales

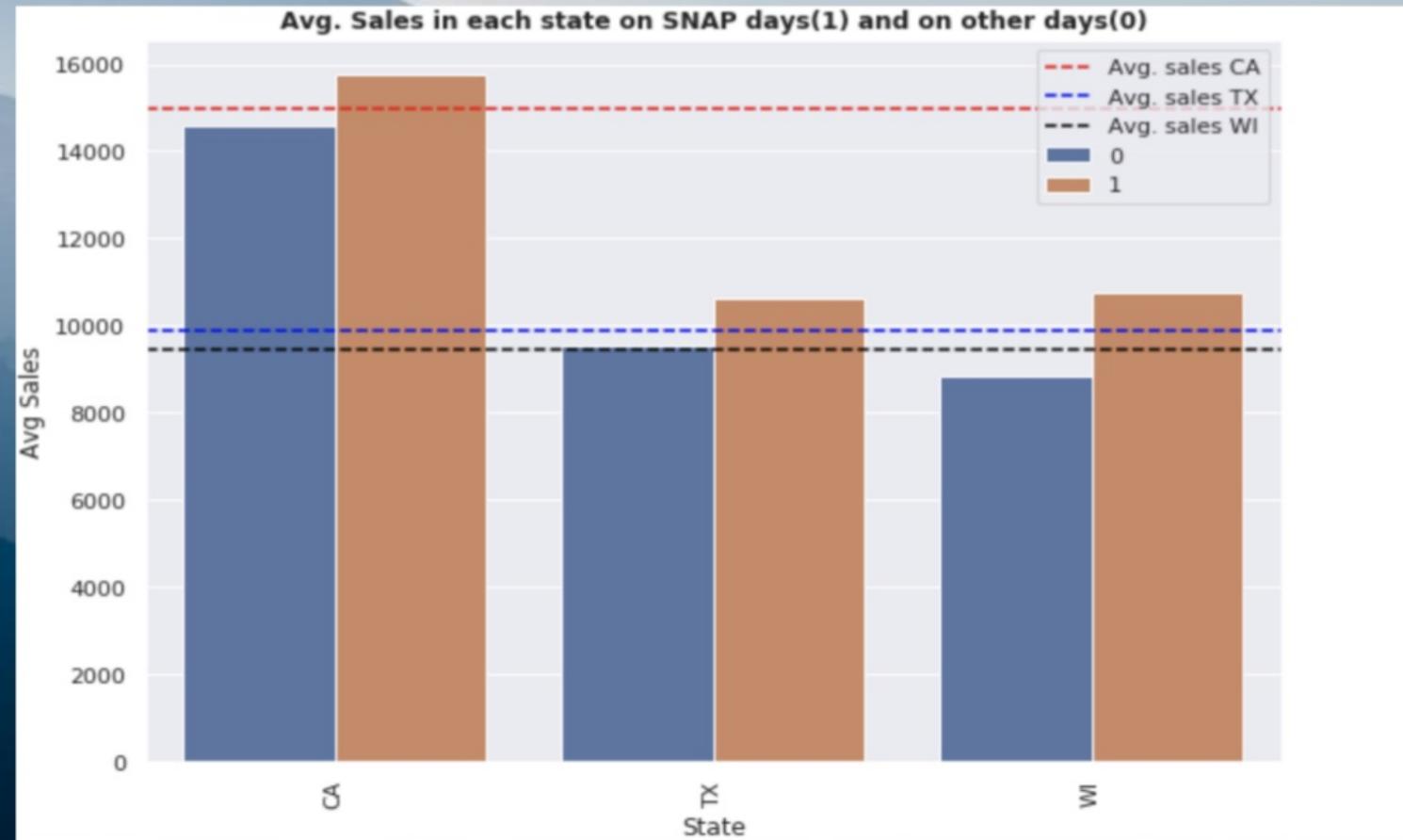


Figure 8: Average sales in each state in SNAP days or not

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.

# EDA: Analysis on Prices Changes



Figure 9: Price and sales changes over years

# Feature Engineering (FE)

- Special Features

- lag\_7: sales shifted 7 steps downwards for each group. The example above focuses on one group only as an example. That is why the first value appears on the 7th index.
- lag\_28: sales shifted 28 steps downwads. That is why the first value appears on the 28th index.
- rmean\_7\_7: rolling mean sales of a window size of 7 over column lag\_7. First value (0.2857) appears on the 13th index because means including nan are nan.
- rmean\_7\_28: rolling mean sales of a window size of 7 over column lag\_28. First value (0.357) appears on the 34th index because that is the first time the mean formula gets all 7 non-nan values.
- rmean\_28\_7: rolling mean sales of a window size of 28 over column lag\_7. First value (0.2857) appears on the 3th index because it is the first time the mean formula gets 28 non-nan values.
- rmean\_28\_28: rolling mean sales of a window size of 28 over column lag\_28. First value appears on 55th index because that is the first time the formula here all non-nan values.

# Feature Engineering (FE)

- Intuition behind the Features

- lag\_7: Captures the week-on-week similarity and that too of just the past week. In other words, people are likely to shop this Monday similar to the last Monday (except it is some special occasion).
- lag\_28: Captures the weekly similarity from a month-to-month perspective. Example: people in the 1st weekend of a month shop more so that weekend looks more similar to first weeks of other months than the previous weekend. (Though 28 is arguable here. A month is generally 30. Interesting would be a variable window depending on when the comparative week starts. Dealing with edge cases like week divided into 2 months will be tricky).
- rmean\_7\_7: Captures the information regarding the sales of the whole previous week ending 7 days in the past i.e. if we are at day 14, then the average is of sales from days 1-7 NOT days 7-14. This provides the information about the whole week and not just a single day sale comparison like lag\_7 to bring the lag\_7 value into "better weekly context".
- rmean\_7\_28: Captures the information regarding the sales of the entire previous 4 weeks ending 7 days in the past i.e. if we are at day 35, then the average is sales from days 1-28.
- rmean\_28\_7: Captures the information regarding the sales of the whole week ending 4 weeks ago i.e. if we are on day 35, then the average is of sales from day 1-7. (Assuming for simplicity the month is 28 days), this provides the information of not just a month-to-month comparison of the same day (day 7 of month one vs day 7 of month two), but the entire week leading up to day 7. Again the idea I believe is to capture the whole week and not just a single day sale comparison like lag\_28 to bring the lag\_28 value into "better weekly context".
- rmean\_28\_28: Captures the information regarding the sales of the entire previous 4 weeks ending 4 weeks in the past i.e. if we are at day 56, then the average is of days 1-28. (Assuming for simplicity the month is 28 days), the idea again is to bring the point value of lag\_28 into a better context (i.e. of day 28 when being compared to day 56) into a "better monthly context".

# LightGBM Model

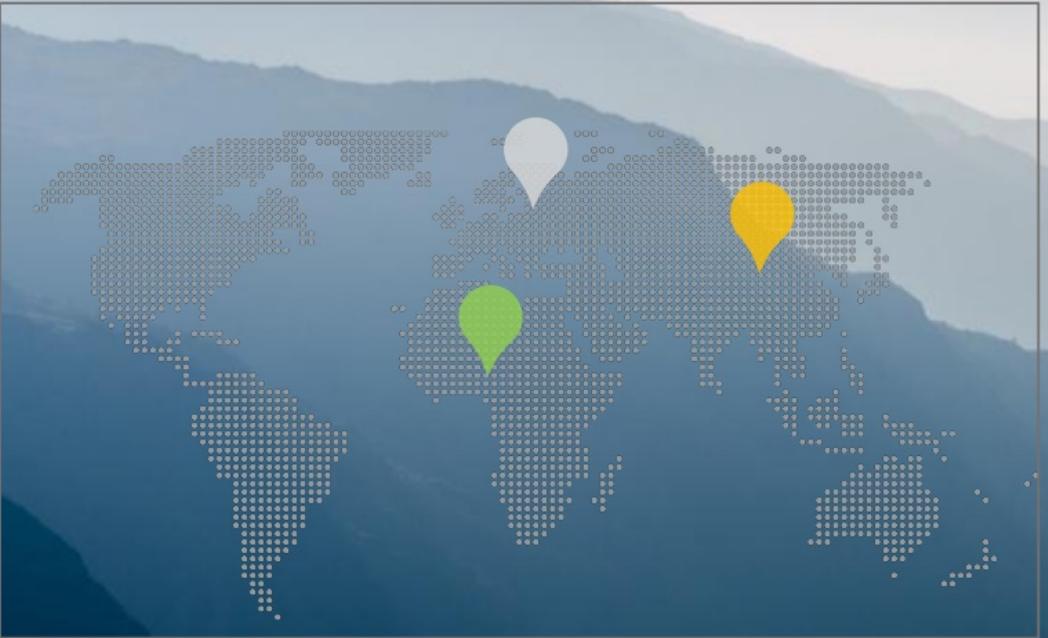
LightGBM is a boosting method, which combines a set of weak learners to form a strong rule.

It is an iterative process.

```
params = {"objective": "poisson", "metric" :"rmse", "force_row_wise" : True,  
"learning_rate": 0.075, "sub_row": 0.75, "bagging_freq": 1, "lambda_l2": 0.1, "metric":  
["rmse"], 'verbosity': 1, 'num_iterations': 1200, 'num_leaves': 128, "min_data_in_leaf":  
100}
```

For feature engineering, the lag feature, rolling feature, and the mean and variance of some statistical features are used, and for saving memory we converting strings to categories, and speed up loading by saving to pickles.

# Forecast



- A recursive way of forecasting
- Trap: when a recursive way is applied on the forecasting, the result from  $t_n$  will be used to predict the next result  $t_{n+1}$ , so it's very easy to understand if you change  $t_1$  by a tiny fraction the data on  $t_{28}$  might be changed a lot.
- Change the weight of the predicting for a little bit and then average them, so get different predictions, and thus to some extent, eliminate the chance we fall into the trap.



# 03

## Performance

|

# Performance

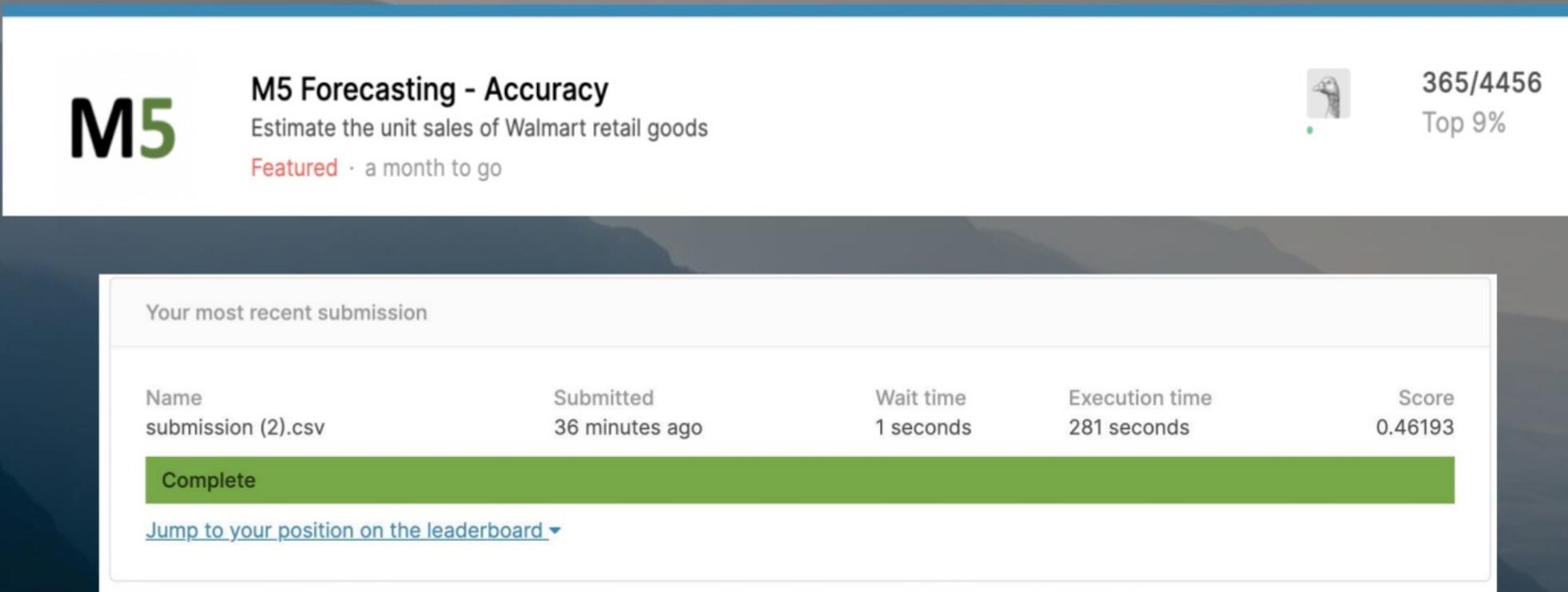


Figure 10: Performance of LightGBM



# 04

## Improvements



## Improvements

When we did EDA, we discovered the point that the impact of SNAP days on sales is not very simple, but linked to the previous weekend, but during FE we failed to produce such a feature to embody the feature.



The loss function in LightGBM is RMSE, which is essentially the same as our target RMSSE. As suggested by some Kagglers, however, if we appropriately choose some other metrics, we might have better performance.

# Work Distribution



- ZHANG Zhiyi: Code Part1 + Write Report
- HU Zhenzhuo: Code Part2 + Write Report
- WU Siliang: Presentation Representative and make the slides.



THANKS

## 2. Slides.pdf

---

### ORIGINALITY REPORT

---

**3%**  
SIMILARITY INDEX

**0%**  
INTERNET SOURCES

**0%**  
PUBLICATIONS

**3%**  
STUDENT PAPERS

---

### PRIMARY SOURCES

---

- |          |   |            |
|----------|---|------------|
| <b>1</b> | <b>Submitted to University of Sheffield</b> | <b>1 %</b> |
|          | Student Paper                               |            |
| <b>2</b> | <b>Submitted to Associatie K.U.Leuven</b>   | <b>1 %</b> |
|          | Student Paper                               |            |
- 

Exclude quotes      Off  
Exclude bibliography      Off

Exclude matches      Off