# MAFS6010Z Artificial Intelligence in Fintech
# Final Project

**TANG Tsz Hong 20735194**

**LAM Chung Wai 20430732**

**CHAN Koon Lam 20748995**

## 1 Kaggle Contest: M5 Forecasting

M5 forecasting competition is the fifth competition held by Makridakis Open Forecasting Center (MOFC) at the University of Nicosia. The aim of the competition is to use the hierarchical sales data from Walmart, estimate the daily sales for the next 28 days (`https://www.kaggle.com/c/m5-forecasting-accuracy`) and estimate its uncertainty distribution. (`https://www.kaggle.com/c/m5-forecasting-uncertainty`).

The following article includes data description, feature engineering, evaluation metrics, model and result. We attempted both Kaggle contest, obtained score of 0.835 and 0.249. The proof is in Appendix II.

## 2 Data Description

Both estimations use the same datasets. The datasets contains the details about sales data of Walmart stores located in the three different US states, California (CA), Texas (TX), and Wisconsin (WI). In the problem, there are three main data sets.

- calendar.csv

  This dataset contains detailed date information of all the 1969 dates, starting from 2011-01-29 to 2016-06-19.

  Important remarks about some of the columns:

  - wday:
    It means weekday. Saturday refers to 1, Sunday refers to 2, etc. The main reason of choosing 1 as Saturday is because the first day in dataset, 2011-01-29, is a Saturday.

  - wm_yr_wk:
    It is id of the week. It corresponds to the last two digits of the year, and the number of week in that year. Notice that first week started on 2011-01-29, start counting the 52 weeks for 2011.

  - snap_CA, snap_TX, and snap_WI
    It is a binary variable. 1 indicates that SNAP purchases are allowed, 0 otherwise. SNAP is a short-form of Supplement Nutrition Assistance Program. It is a nutrition assistance benefit provided by the United States federal government.

- event_type_1 & event_type_2 and event_name_1 & event_name_2:
  It contains the event of that date. The last digit in the column names, 1 and 2, refers to the first event and the second event on that date. For event type, the four types of event are Sporting, Cultural, National, Religious respectively. The following shows some of the event name of each event types.

  Sporting: SuperBowl, NBAFinalsStart, NBAFinalsEnd
  Cultural: Valentines Day, StPatricks Day, Halloween, Easter
  National: Presidents Day, Independence Day, LaborDay
  Religious: LentStart, Purim End, Ramadan starts

- sell_prices.csv

  It is the dataset contain information about price and type of products.

  Important remarks about some of the columns:

  - store_id
    There are 4 stores in California (CA), 3 stores in Texas (TX) and 3 stores in Wisconsin (WI). They are denoted as CA_1, CA_2,...,TX_1,...,WI_3 respectively.

  - sell_price:
    It refers to the price of the product for the given week or store. This price is provided on weekly basis, in other words it takes the average across seven days.

- sales_train_validation.csv / sales_train_evaluation.csv

  After splitting the date by time, it would be the training set, validation set and test set. They contain the time series data that indicate how many items are sold everyday.

  Important remarks about some of the columns:

  - dept_id:
    It means the department of the item came from. There are three major categories Hobbies, Household, and Food. while seven departments, they are HOBBIES_1, HOBBIES_2, HOUSEHOLD_1, HOUSEHOLD_2, FOODS_1, FOODS_2, FOODS_3 respectively.

  - item_id
    it is the id of every unique item. There are 30490 different items in total.

**Combining the datasets**

For further investigations we try to combine the datasets into one main dataframe by the key column, "d". The datasets are merge by the column d, in total there are 59181090 of data. It is then split into training set, validation set, test set with the below time range

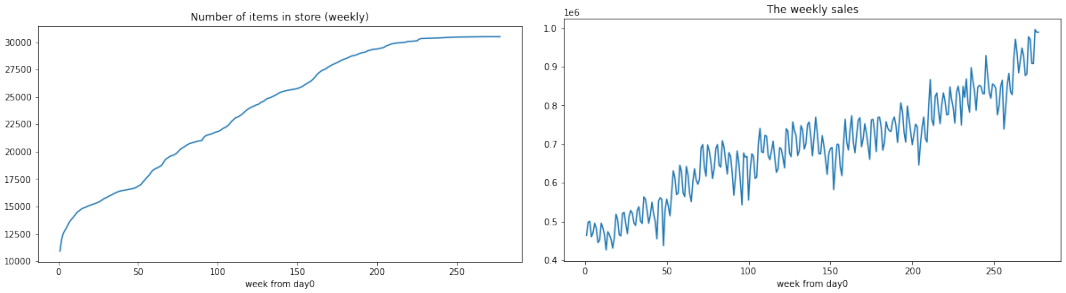
training set: d_1 to d_1913 (2011-01-29 $\sim$ 2016-04-24)
validation set: d_1914 to d_1941 (2016-04-25 $\sim$ 2016-05-22)
test set: d_1942 to d_1969 (2016-05-23 $\sim$ 2016-06-19)

# 3 Feature Engineering

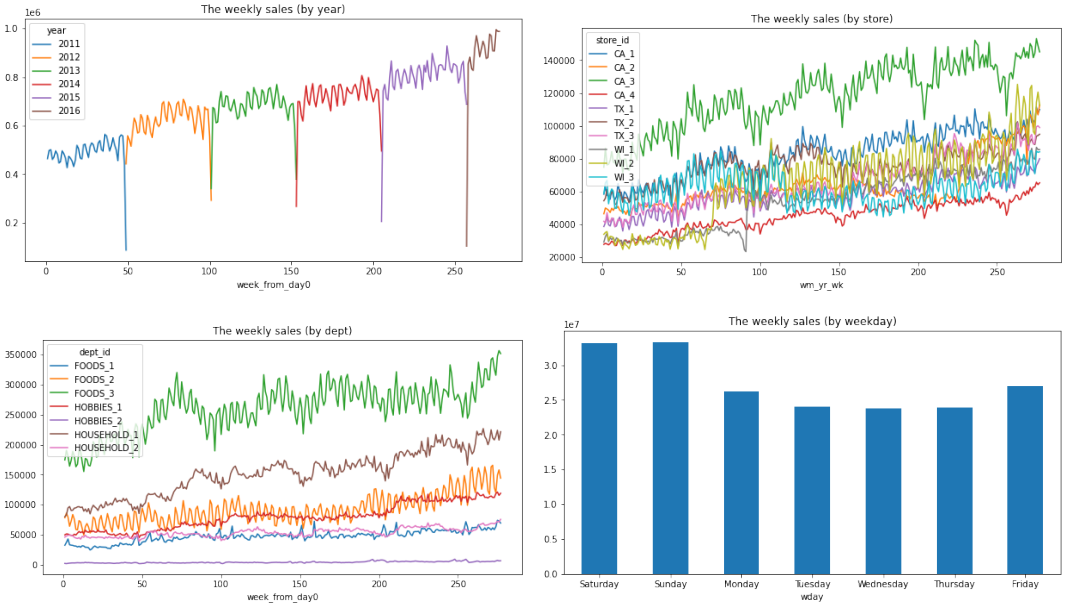
Feature Engineering is an important aspect in machine learning. It refers to the process of using domain knowledge to translate raw data into features, increase the prediction power of the model. For a data scientist, over 60% of time is organising the data, including Feature Engineering.

## 3.1 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is useful to get an overview of how the data behaves. Visualization and statistical techniques will be used in this stage.




The number of item in store keep increasing every week. Customers can reach more variety of items, thus increase their chance of purchasing. Meanwhile, the weekly sales trend also increases by time. It can be explained by inflation, the item price increases, thus lead to the sales increase.




The weekly sales data by store, or by dept, have similar trend of the overall weekly sales data. Also, for the sales revenue by year chart, there are about 12 spikes every year, and their shapes are similar to each other, we suspect that it has some monthly or seasonal trend. The by weekday chart shows the sales amount in weekend is much higher than in weekday.

### 3.2 Adding more features

In this problem, the dataset is complete, so we do not need to deal with missing values. However, the total number of features is too small. Also, based on the previous analysis, we believe that the data has some repetitive trend by time. We also know that U.S. citizens are willing to spend money on holidays, so we added holiday related features e.g. holiday in weekday, christmas etc.

#### Categorical data

In any dataset, there are either numerical data, categorical data, or both. Common ways to handle categorical data are Label encoding and One hot encoding. For example, One hot encoding can be used on column event_name_1, so that the event itself can be analysed individually. We have included the columns christmas, blackfriday, which are the important event for U.S. citizen to celebrate.

#### Lag features

It is also common to add lag features, they are useful in predicting time series data. We have added rolling mean and standard deviation of sales, with lag 7, 30, 60, 90, 180. This indicates weekly lag, monthly lag, bimonthly lag etc.

#### A quick summary of the newly added features

| Type | Column name | dtype |
|---|---|---|
| date related features | year | int |
| | month | int |
| | day | int |
| | week | int |
| | weekend | int |
| | week_from_day0 | int |
| holiday related features | holiday | bool |
| | holiday_in_weekday | bool |
| | christmas | bool |
| | newyearseve | bool |
| | blackfriday | bool |
| | mothersday | bool |
| sales related features | sales | float |
| lag related features | rolling mean of sales with 7, 30, 60, 90, 180 lag | float |
| | rolling std of sales with 7, 30, 60, 90, 180 lag | float |

# 4 Evaluation Metrics

Instead of using the common ways of model evaluation techniques in Machine Learning, like the AUC curve, ROC curve, Confusion Matrix or the Root Mean Square Error (RMSE), the organizer MOFC used other evaluation approaches, WRMSSE and WSPL. This would be the score obtained on Kaggle. Lower score refers to better prediction. Although it is not included in our model training part, it is good to know how the final evaluation works.

## 4.1 Accuracy

The forecast of daily sales for the next 28 days and the forecast of the uncertainty distribution uses different evaluation scheme in Kaggle. For estimating the daily sales, the forecasts will be evaluated using the Root Mean Squared Scaled Error (RMSSE). The formula is as follows:

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

where $n$ denote the length of the training sample, $h$ is the forecasting horizon.

The final Kaggle score is determined by Weighted RMSSE (WRMSSE). Lower WRMSSE score means better performance of the prediction.

## 4.2 Uncertainty

For the second part of the forecasting the uncertainty distribution, it will be evaluated using the Scaled Pinball Loss (SPL) function.

$$SPL(u) = \frac{1}{h} \cdot \frac{\sum_{t=n+1}^{n+h} ((Y_t - Q_t(u))u\mathbb{I}\{Q_t(u) \leq Y_t\} + (Q_t(u) - Y_t)(1-u)\mathbb{I}\{Q_t(u) > Y_t\})}{\frac{1}{n-1}\sum_{t=2}^{n} |Y_t - Y_{t-1}|}$$

where $Q_t(u)$ is the the forecast for quantile u, $n$ denote the length of the training sample, $h$ is the forecasting horizon. The quantile $u$ is set to $u_1 = 0.005, u_2 = 0.025, u_3 = 0.165, u_4 = 0.25, u_5 = 0.5, u_6 = 0.75, u_7 = 0.835, u_8 = 0.975$, and $u_9 = 0.995$.

Again, the final Kaggle score is determined by weighted version of the evaluation metrics, which is the Weighted SPL (WSPL). Lower WSPL score means better performance of the prediction.

# 5 Model and Result

In our analysis, we used Light Gradient Boosting Machine (LightGBM) as the main model. It is a model based on Gradient Boosting Decision Tree (GBDT). This GBDT implementation with Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bunding (EFB) by Microsoft Research team, is the powerful model LightGBM.

## 5.1 Advantages of LightGBM

XGBoost was a popular tool for GBDT before LightGBM announced. XGBoost uses level-wise framework, in other words horizontal growth; while LightGBM uses leaf-wise framework, in other words vertically growth. Therefore the computational cost of LightGBM is much less than XGBoost, however LightGBM is more likely to overfit than the XGBoost. Therefore, the overcome this problem, early stopping criteria is required for LightGBM.

From the original essay of LightGBM, it has the below advantages:

- Faster training speed and higher efficiency
- Lower memory usage
- Better accuracy
- Support of parallel and GPU learning

## 5.2 Methodology

### 5.2.1 Accuracy

For the Accuracy contest, we set the parameter as follows. Rolling window approach is used for cross validation. The boosting type chosen in LightGBM is Gradient Boosting Decision Tree (GBDT). Learning rate is set to be 0.005. To prevent memory issues, we considered force_col_wise to be True. Early stopping criteria is set as 50, to avoid overfitting problem.

### 5.2.2 Uncertainty

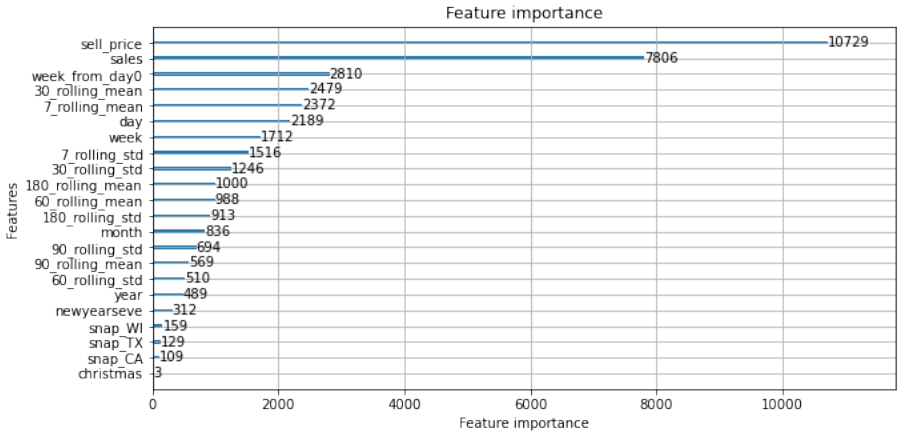
In this contest, Uncertainty prediction means the 28 days ahead probabilistic forecasts for the median and four prediction intervals.

| Level id | Aggregation Level | Number of series |
|---|---|---|
| 1 | Unit sales of all products, aggregated for all stores/states | 1 |
| 2 | Unit sales of all products, aggregated for each State | 3 |
| 3 | Unit sales of all products, aggregated for each store | 10 |
| 4 | Unit sales of all products, aggregated for each category | 3 |
| 5 | Unit sales of all products, aggregated for each department | 7 |
| 6 | Unit sales of all products, aggregated for each State and category | 9 |
| 7 | Unit sales of all products, aggregated for each State and department | 21 |
| 8 | Unit sales of all products, aggregated for each store and category | 30 |
| 9 | Unit sales of all products, aggregated for each store and department | 70 |
| 10 | Unit sales of product *x*, aggregated for all stores/states | 3,049 |
| 11 | Unit sales of product *x*, aggregated for each State | 9,147 |
| 12 | Unit sales of product *x*, aggregated for each store | 30,490 |
| **Total** | | **42,840** |

With the aid of the table provided by MOFC. There are in total 12 aggregation levels that classify the sales data into subgroups.

Our approach is to use normal distribution and the 12 aggregation levels to convert the Accuracy predictions into Uncertainty probabilistic predictions.

### 5.2.3 Feature Importance




The most important features are selling price and sales. Date related features and lag related features are more important than the holiday ones

## 6 Conclusion

After implementing feature engineering and LightGBM model, we obtained the Kaggle score of 0.835 for Accuracy, and 0.249 for Uncertainty, as shown in Appendix II. There are some interesting findings during the project. Before introducing lag features, our score was over 1, while lag features improved the prediction score to 0.84. The improvement is quite significant.

For future improvements, when we compare others work on Kaggle, some of the models take more than 10 hours training time or even one day to process while ours only take about 1 hour. More complex model may be one of the ways yielding better predictions. Also, we could add more features, for example, price lag, maximum, minimum, norm.

# Appendix

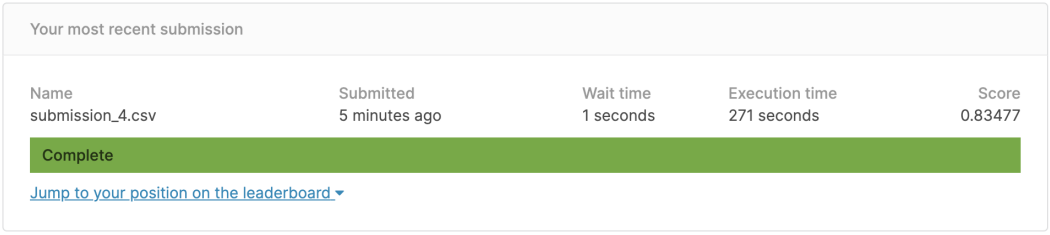
## Appendix I : Individual's responsibilities

Table 1: Contribution of groupmates

| Name | Student ID | Contribution | Duty |
|---|---|---|---|
| TANG Tsz Hong | 20735194 | 33.33% | Feature Engineering, Modelling, Writing report, Presentation |
| LAM Chung Wai | 20430732 | 33.33% | Feature Engineering, Modelling, Writing report, Presentation |
| CHAN Koon Lam | 20748995 | 33.33% | Feature Engineering, Modelling, Writing report, Presentation |

## Appendix II : Kaggle score

Score of M5-forecasting-accuracy




Score of M5-forecasting-uncertainty