Predictive Analytics Team Project Write Up
M5 Forecasting : Accuracy

Chen Ju Wang, Shang Chien Wang, Ya Chin Hsu, Yi Hsiang Yen

Table 1: Explored Models, Hyper-Parameters

Model	Feature	Value
LightGBM	- Objective - Num_leaves - N_estimators - Learning_rate	- Regression - 64 - 100 - 1
LSTM	- Hidden dimension - Number of LSTM layers - Final output layer - Loss function - Optimizer - Learning rate - Batch size - Number of epochs - Input sequence length - Prediction horizon	- 128 - 2 - Fully connected (Linear) - Mean Squared Error (MSE) - Adam optimizer - 0.001 - 32 - 10 - 28 - 56
GRU	- GRU hidden units - Input sequence length - Features - Dropout rate - Dense layer units - Batch_size - Maximum epochs - Steps_per_epoch - Adam optimizer learning rate - Early stopping patience	- 64 - 28 - 26 - 0.2 - 32 - 16 - 20 - Dynamically - 0.0001 - 5 epochs

Table 2: Explored Features, Rationale of Feature Engineering

Model		Feature	Feature Engineering Rationale
Tree-based model	Geographic/Product-related	State_id	categorical column indicating the U.S. state of each store, pulled directly from the state_id field in sales.csv.
		Store_id	categorical column indicating the unique store identifier, pulled directly from the store_id field in sales.csv.
		Cat_id (Only in LSTM/GRU)	categorical column indicating the product category, pulled directly from the cat_id field in sales.csv and fed into the GRU as an embedding

			input.
		Dept_id	categorical column indicating the product department, pulled directly from the dept_id field in sales.csv.
	Calendar-related	Wday	integer column indicating day of week (1–7), taken from the wday field in calendar.csv.
		Is_weekend	binary column indicating weekends, engineered by setting to 1 when wday ∈ {6,7} and 0 otherwise using calendar.csv.
		Month	integer column indicating calendar month (1–12), taken from the month field in calendar.csv.
		Year	integer column indicating calendar year (e.g. 2011–2016), taken from the year field in calendar.csv.
		Is_event (Only in LightGBM)	binary column indicating special event days, engineered by setting to 1 if either event_name_1 or event_name_2 is non-null in calendar.csv.
		snap_CA, snap_TX, snap_WI	binary columns indicating SNAP benefit distribution days in California, Texas, and Wisconsin, pulled directly from the snap_CA, snap_TX, snap_WI fields in calendar.csv.
		Wm_yr_wk	integer column indicating year-week key, taken from the wm_yr_wk field in calendar.csv to align daily sales with weekly prices.
		Event_type (Only in Neural Network Models)	categorical column indicating the type of special event, taken from event_type_1/event_type_2 in calendar.csv, then one-hot encoded or embedded for the neural network.
	Price-related	Price	double column indicating unit sale price, pulled from the sell_price field in sell_prices.csv
		Max_price	double column indicating historical maximum price, engineered by taking the maximum of price over all past records for each (item_id, store_id).
		Min_price	double column indicating historical minimum price, engineered by taking the minimum of price over all past records for each (item_id, store_id)
		Mean_price	double column indicating historical average price, engineered by taking the average of price over all past records for each (item_id, store_id)
		Stddev_price	double column indicating price volatility, engineered by computing the sample standard deviation of price over all past records for each (item_id, store_id).
		D28_moving_avg_price (Only in LightGBM)	double column indicating 28-day rolling average price, engineered by averaging price over the preceding 28 days per item_id, falling back to the cumulative average when fewer than 28 days exist.
		Lag1_price (Only in LightGBM)	double column indicating previous day’s price, engineered by shifting price by one day within each item_id and back-filling nulls with the current price.

		Price_diff_value (Only in LightGBM)	double column indicating absolute price change, engineered by calculating price – lag1_price for each day per item_id.
		Price_diff_percentage (Only in LightGBM)	double column indicating relative price change (%), engineered by (price – lag1_price) / lag1_price * 100, with zeros for cases where lag1_price is zero.
	Sales-related	Max_sales	double column indicating historical peak sales, engineered by taking the maximum of daily sales over all past records for each item_id.
		min_sales	double column indicating historical lowest sales, engineered by taking the minimum of daily sales over all past records for each (item_id, store_id).
		Mean_sales	double column indicating historical average sales, engineered by taking the average of daily sales over all past records for each (item_id, store_id).
		Stddev_sales	double column indicating sales volatility, engineered by computing the sample standard deviation of daily sales over all past records for each (item_id, store_id).
		D28_moving_avg_sales (Only in LightGBM)	double column indicating 28-day rolling average sales, engineered by averaging sales over the preceding 28 days per (item_id, store_id), falling back to the cumulative average when fewer than 28 days exist.
		Lag1_sales (Only in LightGBM)	double column indicating previous day's sales, engineered by shifting sales by one day within each (item_id, store_id) and filling early nulls with the item-store's historical mean.

Table 3: Model Performance Comparison

- LSTM（**Best Performance Model**）

Private Score	Public Score
1.50406	1.37348

- GRU

Private Score	Public Score
3.24086	5.44561