**Table 1: Explored Models & Hyperparameters**

| Model Type | Key Hyperparameters |
|---|---|
| Linear Regression | n_jobs=7 |
| Adaboost | n_estimators=5,learning_rate=0.1,loss='square' |
| XGBoost Regressor | n_estimators=100, max_depth=6, learning_rate=0.1, subsample=0.8, colsample_bytree=0.8, tree_method='hist' |
| LightGBM | params = {"objective": "tweedie",<br>"metric": "rmse",<br>"tweedie_variance_power": trial.suggest_float("tweedie_variance_power", 1.05, 1.3),<br>"learning_rate": trial.suggest_float("learning_rate", 0.01, 0.05),<br>"num_leaves": trial.suggest_int("num_leaves", 128, 512),<br>"min_data_in_leaf": trial.suggest_int("min_data_in_leaf", 256, 2047),<br>"feature_fraction": trial.suggest_float("feature_fraction", 0.6, 0.95),<br>"bagging_fraction": trial.suggest_float("bagging_fraction", 0.6, 0.95),<br>"bagging_freq": 1,<br>"lambda_l2": trial.suggest_float("lambda_l2", 0.0, 1.0),<br>"boost_from_average": True,<br>"max_bin": 127,<br>"bin_construct_sample_cnt": 20000000,<br>"force_row_wise": True,<br>"verbosity": 1,<br>"nthread": 8,<br>"seed": 42} |
| LSTM (Encoder) | **Embedding + LSTM** structure for categorical inputs<br>**LSTM** for sequential numerical features<br>**Dense layers with varied activation functions**<br>**Batch normalization** for stabilization<br>**Activations**: Mix of relu, sigmoid, tanh across layers |
| CNN LSTM | **Embedding dimensions** range from 10 to 150<br>**Conv1D filters** range from 3 to 7; **kernel size** = 8 (categorical), 16 (numerical)<br>**LSTM units**: 50 (Item_Id), 10 (others), 32 (numerical)<br>**Dense layers**: 256 → 128 → 64 → 1<br>**Batch normalization** for stabilization<br>**Activations**: Mix of relu, sigmoid, tanh across layers |

**Table 2: Engineered Features & Rationale**

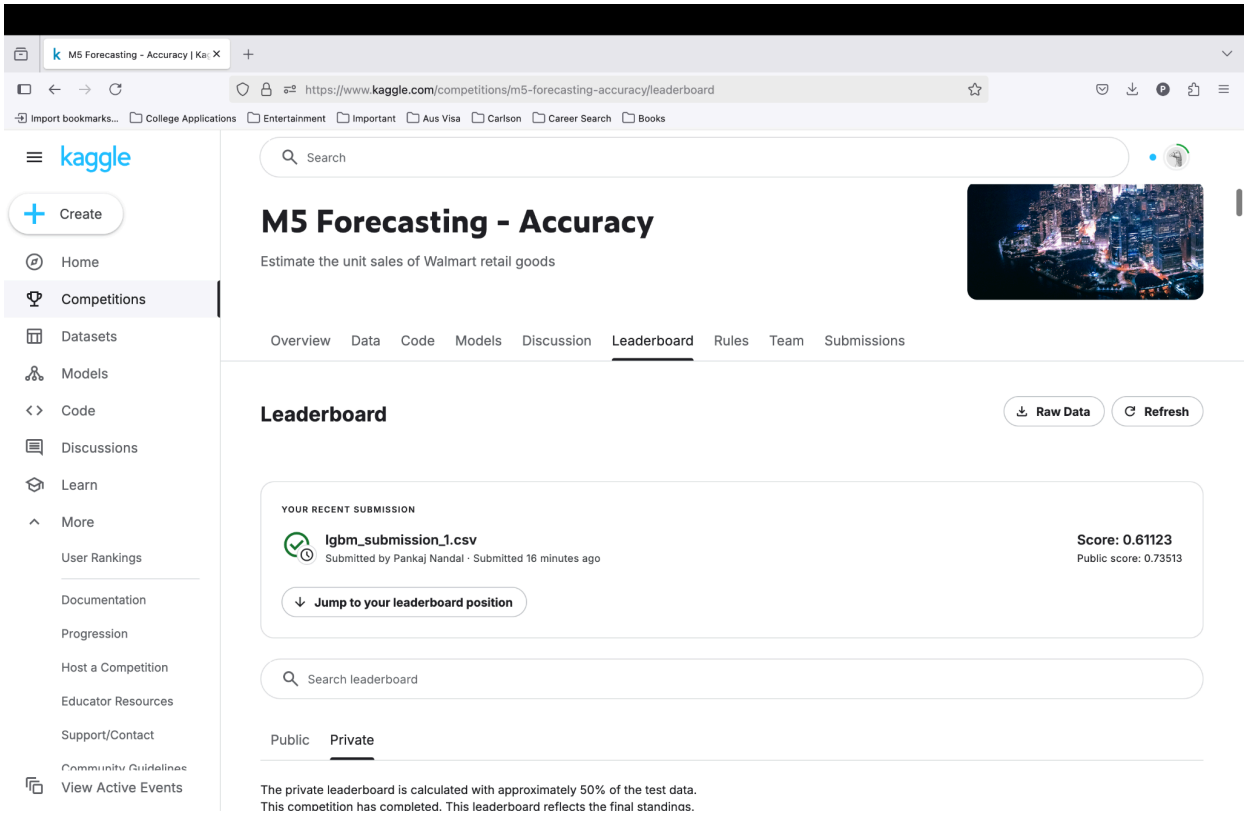
| Type | Feature | Description | Column_Name | Why do we use | Used in Final Model |
|---|---|---|---|---|---|
| **Categorical Encoding** | item_id | Item ID | item_id | Encoding transforms categorical features into **model-readable, numerical formats**. | Yes |
| | dept_id | Department ID | dept_id | | Yes |
| | cat_id | Category ID | cat_id | | Yes |
| | store_id | Store ID | store_id | | Yes |
| | state_id | State ID | state_id | | Yes |
| | event_name_1 | Event Name | event_name_1 | | Yes |
| | event_name_2 | Event Name | event_name_2 | | Yes |
| | event_type_1 | Event Type | event_type_1 | | Yes |
| | event_type_2 | Event Type | event_type_2 | | Yes |
| | year | Year of sale | year | | Yes |
| **Calendar Features** | dayofweek | Day of week (1=Saturday) | wday | Helps model seasonality, weekly and monthly cycles. | Yes |
| | weekofyear | Week number (ISO Week Number?) | weekofyear | | Yes |
| | month | Month number | month | | |
| | year | Year number | year | | Yes |
| | day | Day of month | day | | |
| | weekend | Flag: is it the weekend? | weekend | | |
| | Is month start / end | If month start or end | is_month_start/end | | Yes |
| | Is quarter start / end | If quarter start or end | is_quarter_start/end | | Yes |
| | Is Year start/end | If year start or end | year_start/year_end | | Yes |
| | Season | Season (WInter, Spring, etc) | season | | Yes |
| | Day of year | Day of the year | day_of_year | | |

| | | | | | |
|---|---|---|---|---|---|
| **Event/Holiday Features** | *is_event* | *Flag: Is there an event?* | *is_event* | Strong lift around holidays and events. | |
| | Event proximity | Days until / since next event | days_since_event/days_until_event | | |
| **Promotion/Snap Features** | snap_CA, snap_TX, snap_WI | SNAP active flags | snap_active | **SNAP (Supplemental Nutrition Assistance Program)** is a U.S. government program that helps low-income individuals and families buy food. | Yes |
| | Days since SNAP active | For SNAP customers, since last promotion active | days_since_snap | | |
| | Days until next SNAP | For SNAP customers, until next promotion active | days_until_next_snap | | |
| **Lag Features** | sales_lag_1 | Yesterday's sales | sales_lag_1 | Time-series models must have lag features to capture autocorrelation. | |
| | sales_lag_7 | Last week's sales | sales_lag_7 | | |
| | sales_lag_14 | Two weeks | sales_lag_14 | | |
| | Sales_lag_28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98 | Custom lag | direct_lag_n | | Yes |
| **Rolling Window Features** | rolling_mean_7/14 | Moving average | rolling_mean_7/rolling_mean_14 | Capture trends and seasonality patterns dynamically. | |
| | rolling_std_7/14 | Rolling standard deviation (volatility) | rolling_std_7/rolling_std_14 | | |
| | expanding_mean | Expanding mean for trend, It's the **cumulative average** of sales up to that point in time (excluding today), calculated per item-store (id). | expanding_mean | | |
| | rolling_min / max | Rolling min/max for local peak detection. It tells you the **lowest sales value** in the **last 7 days** before a given day. | rolling_min_7/rolling_max_7 | | |
| | Rolling statistical mean of 7, 14, 30, 60, 360 days with shift of 28 days | **Rolling statistical features** (mean) over past sales data, with a **shift of 28 days** to **prevent data leakage**. | roll_n_shift_28_mean | | Yes |
| | Rolling statistical standard deviation of 7, 14, 30, 60, 360 days with shift of 28 days | **Rolling statistical features** (standard deviation) over past sales data, with a **shift of 28 days** to **prevent data leakage**. | roll_n_shift_28_std | | Yes |
| **Cumulative /Expanding Features** | cumulative_sales | Cumulative sum of sales | cumulative_sales | Good for lifecycle analysis and stockout behavior. | |
| | cumulative_mean_sales | Expanding mean of sales | cumulative_mean_sales | | |

| | | | | | |
|---|---|---|---|---|---|
| | days_since_last_sale | Counter since last sale event (in case of NA counted days since start) | days_since_last_sale | | |
| **Hierarchical Aggregated Features** | sales at dept/store level | Aggregated sales at dept or store level | dept_sales/store_sales | Aggregations help capture group-level seasonality and shared patterns. | |
| | sales at state/category level | Aggregated sales at state or category level | state_sales/cat_sales | | |
| | rolling mean at group level | Rolling means for groupings (store-dept) | rolling_mean_store_dept_7/ rolling_mean_store_dept_7_lag_7 | | |
| **Lag on Engineered Features** | Lag of rolling mean | Lag of moving averages | rolling_mean_7_lag_7/ rolling_mean_14_lag_14 | Captures Temporal Patterns in Aggregates. | |
| | Rolling Mean Change | Derive Growth Rates | rolling_mean_change | | |

**Table 3: Model Performance Comparison**

| Model | Private WRASSE |
|---|---|
| Linear Regression | 0.883 |
| Adaboost | 1.094 |
| XGBoost | 2.12 |
| LightGBM | 0.611 |
| LSTM (Encoder) | 0.980 |
| CNN LSTM | 1.285 |
| Ensemble | 0.616 |


M5 Forecasting - Accuracy | Ka
https://www.kaggle.com/competitions/m5-forecasting-accuracy/leaderboard
Import bookmarks...
College Applications
Entertainment
Important
Aus Visa
Carlson
Career Search
Books
kaggle
Search
Create
Home
Competitions
Datasets
Models
Code
Discussions
Learn
More
User Rankings
Documentation
Progression
Host a Competition
Educator Resources
Support/Contact
View Active Events
M5 Forecasting - Accuracy
Estimate the unit sales of Walmart retail goods
Overview
Data
Code
Models
Discussion
Leaderboard
Rules
Team
Submissions
Leaderboard
Raw Data
Refresh
YOUR RECENT SUBMISSION
lgbm_submission_1.csv
Submitted by Pankaj Nandal · Submitted 16 minutes ago
Score: 0.61123
Public score: 0.73513
Jump to your leaderboard position
Search leaderboard
Public
Private
The private leaderboard is calculated with approximately 50% of the test data.
This competition has completed. This leaderboard reflects the final standings.