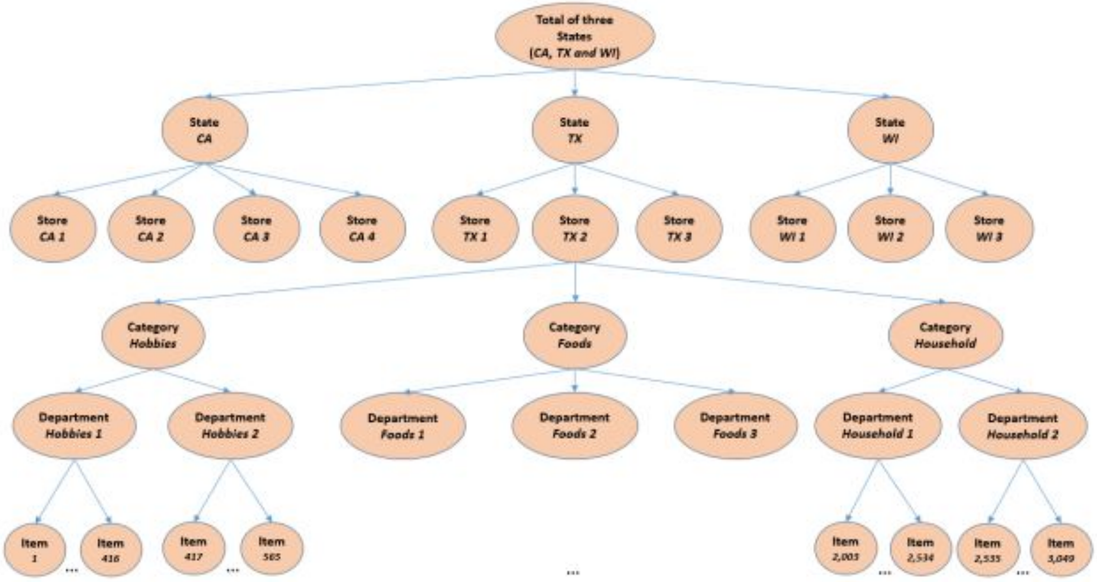
Product Sales Forecasting Solution

Jacob Battles, Ko-Jen Kang, Pin-Shiuan Liang, Tsai-Ning Lin, Vikhyat Tomar

Our Goal

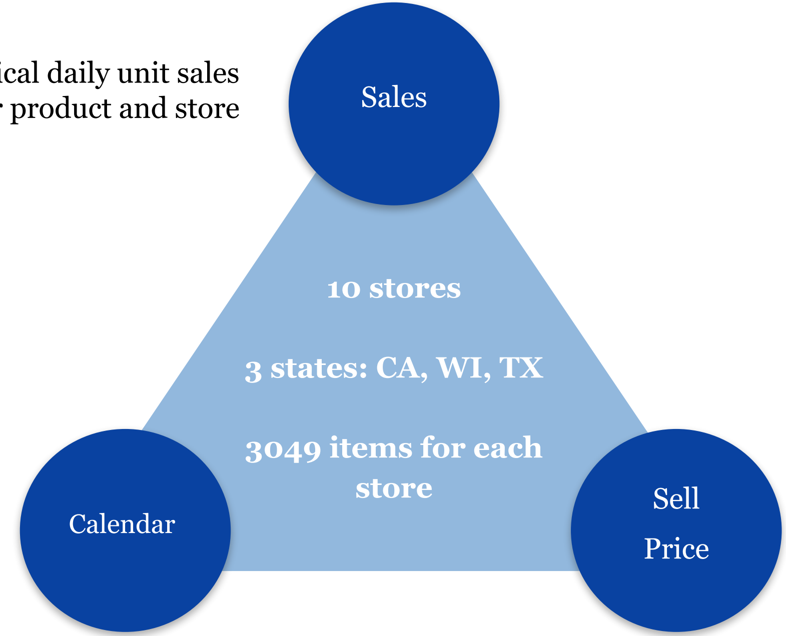
Predict item sales at stores in various locations for 28-day time periods

Used bottom-up solution, getting the best predictive model with lowest Weighted Root Mean Squared Scaled Error (WRMSSE)



Data Source

Contains the historical daily unit sales data per product and store



Contains information about the dates the products are sold

Contains information about the price of the products are sold per store and date

Process Map

1

Data Consolidation

Merge 3 data sources into

1 combined file

2

Feature engineering

Generate additional

meaningful features

3

Data Preprocessing

- Normalization
- Handling missing values

4

Model training

(LightGBM/ Sequence2Sequence)

5

Generate predictions

Predict future 28 days sales

for each item in each store

Rationale of Feature Engineering

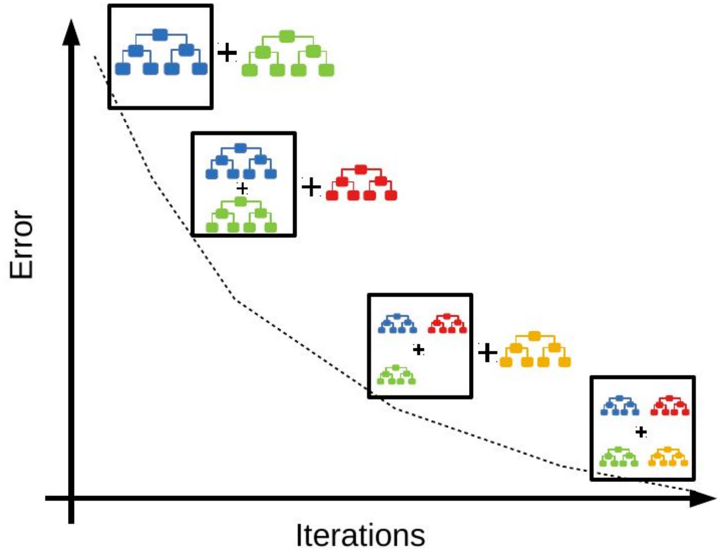
Category	Rationale	Features
Lag sales features	Capture **sales patterns from previous time points** to model seasonality and trends	lag_7, lag_28
Rolling sales mean	**Smooth short-term fluctuations** and highlight underlying sales trends	rolling_mean_7, rolling_mean_28
Calendar features	Account for **time-based effects** and recurring patterns in sales	dayofweek, month, year, is_weekend, days since the product being sold
Event flag	Capture **demand spikes** caused by special events or promotions	is_event
Price features	Reflect how **price changes or discounts** influence sales	price_change_rate, price_event_interaction

Model Exploration

Single LightGBM

LightGBM is a fast, smart machine learning tool that makes predictions from data.

- Builds **lots of small decision trees**
- Learns from mistakes to improve accuracy
- Great for **large datasets** and **quick results**



Example:
Predicting house prices based on size, location, and number of rooms.

Using one single LightGBM Model gives us a good baseline to compare against when building more complex models.

Model Comparison

Model	Model Performance (WRMSSE)	Logic	Training data
Single LightGBM (Recursive)	0.71393 (Regression, no tuning) 0.72501 (Optuna Tuning) 0.79857 (Tweedie, no tuning)	Tuned using Optuna. Recursive: Uses all available history, including predictions, to predict future days	d_1~d_1913 as training data d_1914~d_1941 as validation data Predict d_1942~d_1969
Ensembled LightGBM (60% Recursive + 40% Non-recursive)	1.00917	Two models per store: (1) Recursive (day-by-day) prediction (2) Non-recursive (one-shot) prediction	d_1~d_1913 as training data d_1914~d_1941 as validation data Predict d_1942~d_1969
Sequence-to-sequence (LSTM for both encoder & decoder)	4.07210	Using past **two years** to predict the following 28 days	d_423 ~ d_1153 as input for encoder d_1154 ~ d_1182 as target for decoder d_1211~ d_1941 as input to predict d_1942~d_1969

Ensembled LightGBM

Why "Ensembled"?

We applied weighted average **(60% recursive + 40% non-recursive)**

Recursive

Predicts day-by-day recursively
(yesterday → today)

+

Non-recursive

Predicts all 28 days in a single shot
(direct prediction)

=

- Combine the strengths of both models
- Balance both long and short term accuracy

Pros: captures daily sales patterns
Cons: may accumulate errors

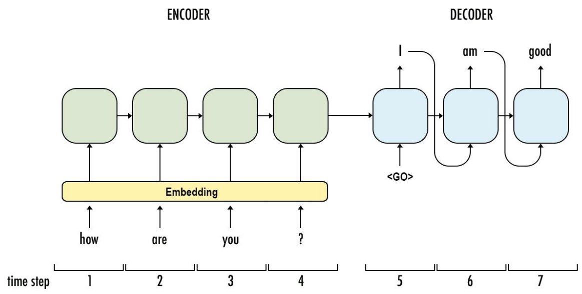
Pros: offers a more stable, overall forecast
Cons: may accumulate errors

Model Comparison

Model	Model Performance (WRMSSE)	Logic	Training data
Single LightGBM (Recursive)	0.71393 (Regression, no tuning) 0.72501 (Optuna Tuning) 0.79857 (Tweedie, no tuning)	Tuned using Optuna. Recursive: Uses all available history, including predictions, to predict future days	d_1~d_1913 as training data d_1914~d_1941 as validation data Predict d_1942~d_1969
Ensembled LightGBM (60% Recursive + 40% Non-recursive)	1.00917	Two models per store: (1) Recursive (day-by-day) prediction (2) Non-recursive (one-shot) prediction	d_1~d_1913 as training data d_1914~d_1941 as validation data Predict d_1942~d_1969
Sequence-to-sequence (LSTM for both encoder & decoder)	4.07210	Using past **two years** to predict the following 28 days	d_423 ~ d_1153 as input for encoder d_1154 ~ d_1182 as target for decoder d_1211~ d_1941 as input to predict d_1942~d_1969

Sequence-to-sequence

What is Seq-2-Sequence?



A Sequence-to-Sequence model (Seq2Seq) is a type of deep learning model that:

- Takes a **sequence of past events** as input (In our case past 2 years of sales)
- Outputs a **sequence of future** predictions (Next 28 days of sales)

This makes it perfect for **time-series** forecasting - like predicting future product demand based on historical trends.

How does it work?

1. Encoder: It looks at the past data (Sales) and summarizes all the important patterns into a compressed "memory" called a context vector.

2. Decoder: It takes that memory and generates the forecast, one day at a time, learning from the context built by the encoder.

3. Prediction Phase: It uses recursive prediction, meaning each predicted value is fed back into the model to generate the next day's forecast — repeated for all 28 days.

Together, they form a pipeline:

Past sales ➜ Encoder ➜ Context ➜ Decoder ➜ Future sales

Model Comparison

Model	Model Performance (WRMSSE)	Logic	Training data
Single LightGBM (Recursive)	0.71393 (Regression, no tuning) 0.72501 (Optuna Tuning) 0.79857 (Tweedie, no tuning)	Tuned using Optuna. Recursive: Uses all available history, including predictions, to predict future days	d_1~d_1913 as training data d_1914~d_1941 as validation data Predict d_1942~d_1969
Ensembled LightGBM (60% Recursive + 40% Non-recursive)	1.00917	Two models per store: (1) Recursive (day-by-day) prediction (2) Non-recursive (one-shot) prediction	d_1~d_1913 as training data d_1914~d_1941 as validation data Predict d_1942~d_1969
Sequence-to-sequence (LSTM for both encoder & decoder)	4.07210	Using past **two years** to predict the following 28 days	d_423 ~ d_1153 as input for encoder d_1154 ~ d_1182 as target for decoder d_1211~ d_1941 as input to predict d_1942~d_1969

Model Comparison

Model	Model Performance (WRMSSE)	Logic	Training data
Single LightGBM (Recursive)	0.71393 (Regression, no tuning) 0.72501 (Optuna Tuning) 0.79857 (Tweedie, no tuning)	Tuned using Optuna. Recursive: Uses all available history, including predictions, to predict future days	d_1~d_1913 as training data d_1914~d_1941 as validation data Predict d_1942~d_1969
Ensembled LightGBM (60% Recursive + 40% Non-recursive)	1.00917	Two models per store: (1) Recursive (day-by-day) prediction (2) Non-recursive (one-shot) prediction	d_1~d_1913 as training data d_1914~d_1941 as validation data Predict d_1942~d_1969
Sequence-to-sequence (LSTM for both encoder & decoder)	4.07210	Using past **two years** to predict the following 28 days	d_423 ~ d_1153 as input for encoder d_1154 ~ d_1182 as target for decoder d_1211~ d_1941 as input to predict d_1942~d_1969

Model Comparison

Model	Advantage	Disadvantage
Single LightGBM	• Fast and lightweight • Handles tabular features well	• Recursive error accumulation • Ignores temporal correlation explicitly • Less robust to long horizons
Ensembled LightGBM	• Balances short- and long-term accuracy • Robust to different time horizons	• More complex to manage • Longer training time
Sequence-to-sequence	• Captures temporal dynamics • Multi-step forecasting in one pass	• Computationally expensive • Hard to interpret • Training instability • Recursive error accumulation

Summary

Simple LightGBM achieved the best performance/efficiency

- From simple baselines to complex architectures, we found that simple LightGBM generalized the best in the unseen data.
- LightGBM also provides fast, efficient, and accurate predictions for large-scale data.

Next steps

- Enhance accuracy by adding more features.
- Deploy the LightGBM model to support demand forecasting.
- Monitor performance and update the model as new data comes in.
- Share forecasts through dashboards to guide business decisions.

Thank you for listening!