FINAL PRESENTATION

Team6

Outline

1. Data Overview
2. Data Preprocessing
3. The Models
4. Result

Data Overview

The Features

Numerical

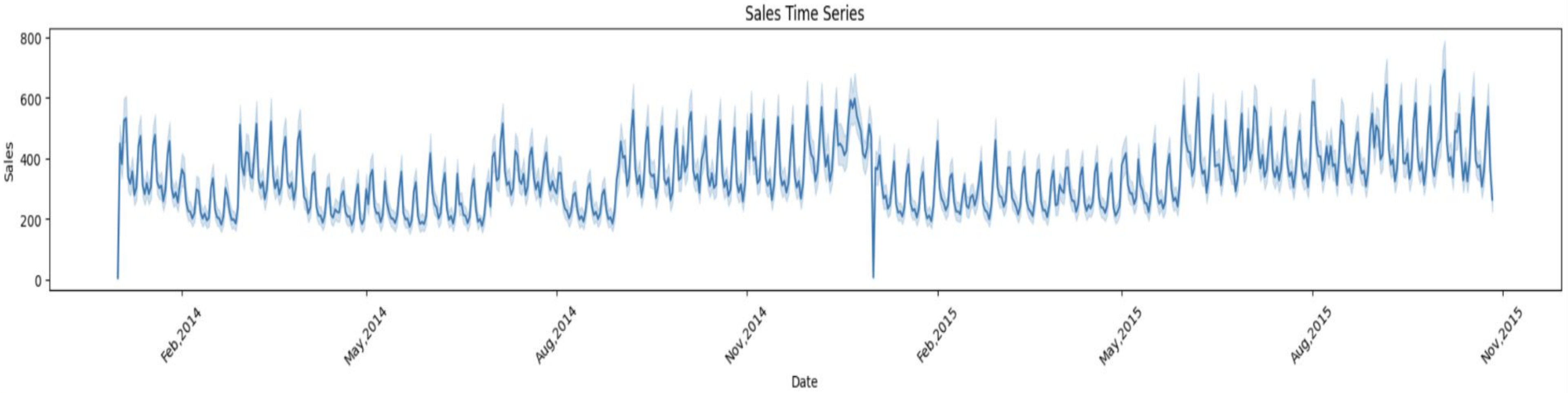
- ❑ On_promotion
- ❑ Oil price

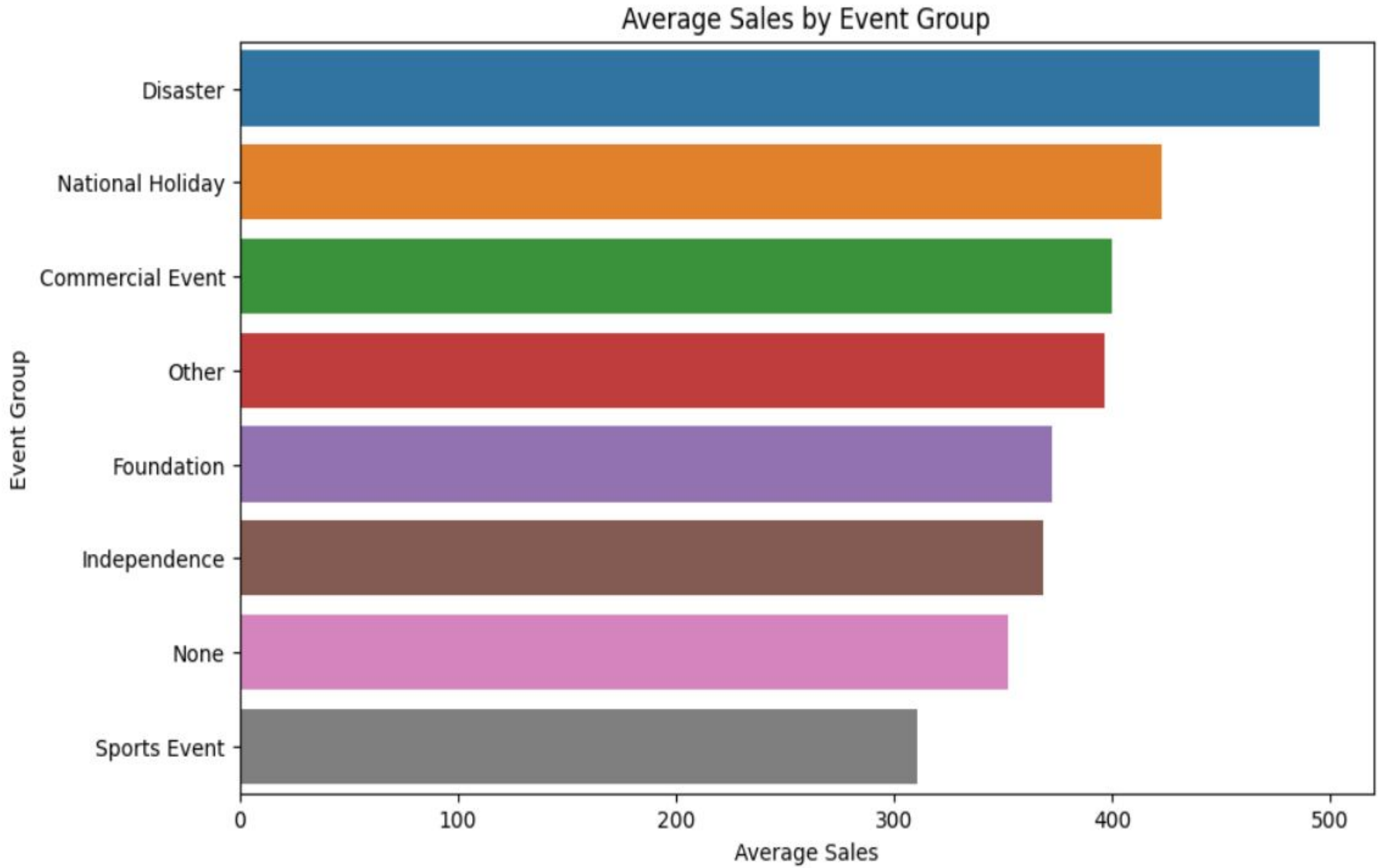
Categorical

- ❑ Store_nbr
- ❑ Family
- ❑ Store type
- ❑ Earthquake
- ❑ Holidays
- ❑ City

Target Feature

- Sales




Average Sales by Event Group
Disaster
National Holiday
Commercial Event
Other
Foundation
Independence
None
Sports Event
Event Group
0
100
200
300
400
500
Average Sales

Problem

- Predict sales for each store and product category in the retail industry.

- Accurate sales forecasting allows for optimization of inventory management and promotional strategies.

Data Preprocessing
and
Feature Engineering

Improvements in Our Approach

- Reduced the number of features
- Created new, effective features

What we used to use:

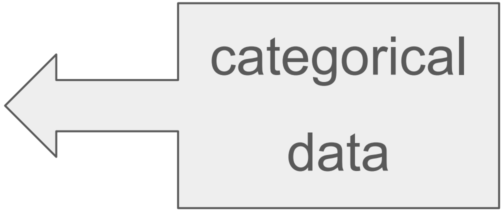
- Store_nbr
- Family
- Day_of_week
- Year
- Onpromotion
- Lags
- Dcoilwtico
- Store
- Event
- City
- Day
- Quarter
- Day_of_year

What we use now:

- Store_nbr
- Family
- Day_of_week
- Year
- Onpromotion
- Lags

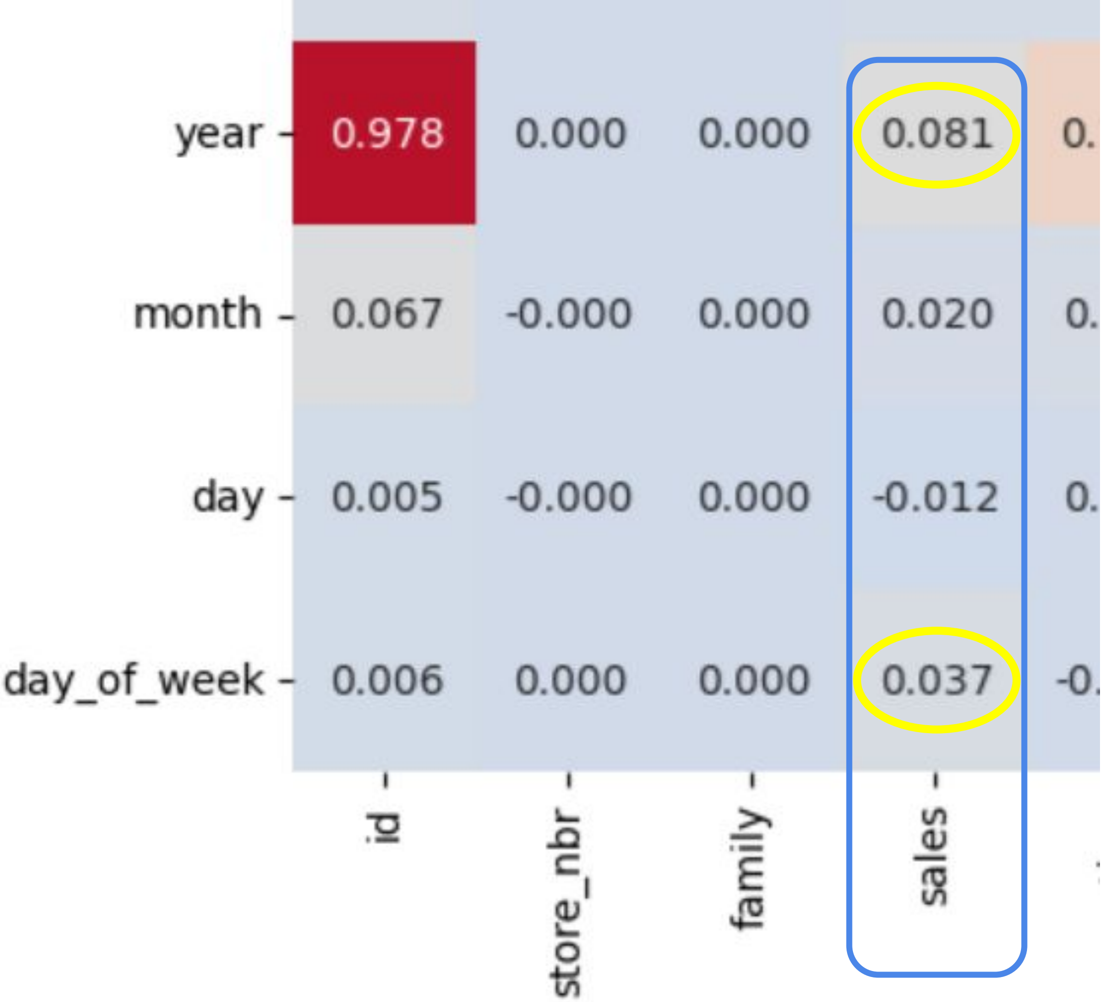
Specific Improvements

- Removed unnecessary categorical columns
- Used LightGBM for efficient handling of categorical data



Simplified Date Features

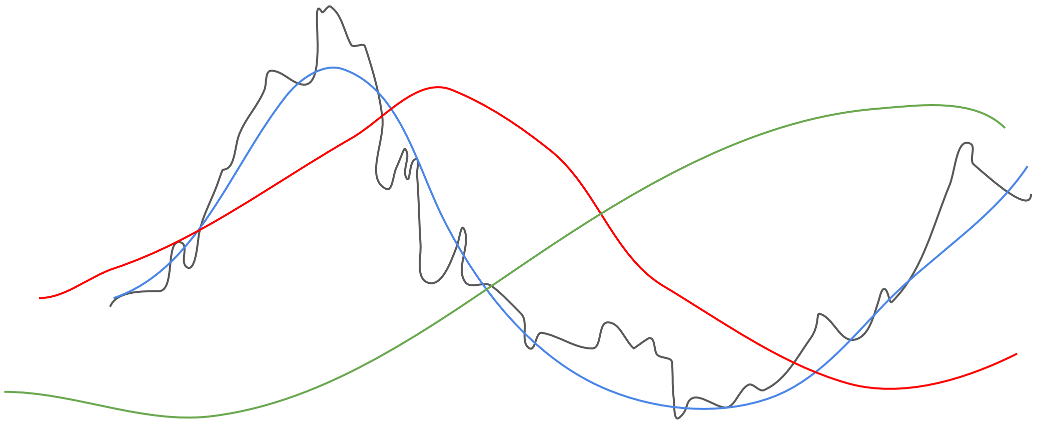
- Focused on two features: day_of_week and year
- Sufficient for effective model learning



More relevant than other date data.

Enhanced Lag Features

- Used multiple lags: 1 day, 7 days, 30 days
- Applied Exponentially Weighted Moving Average (EWMA)
- Captured sales trends more accurately



Data Combination and Cleaning

- Combined training and test data for consistent feature engineering
- Cleaned data to remove noise

Conclusion

- Reduced features to simplify the model
- Prevented overfitting

The Models

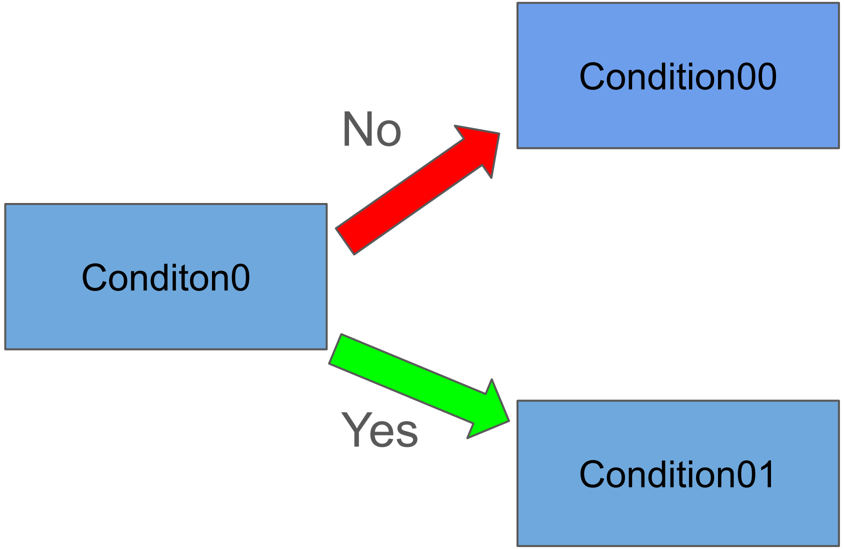
Things to keep in mind

- Handling discrete variables
- The target variable is continuous
- The data is a time series

The Basic : Decision Tree

Core characteristic :

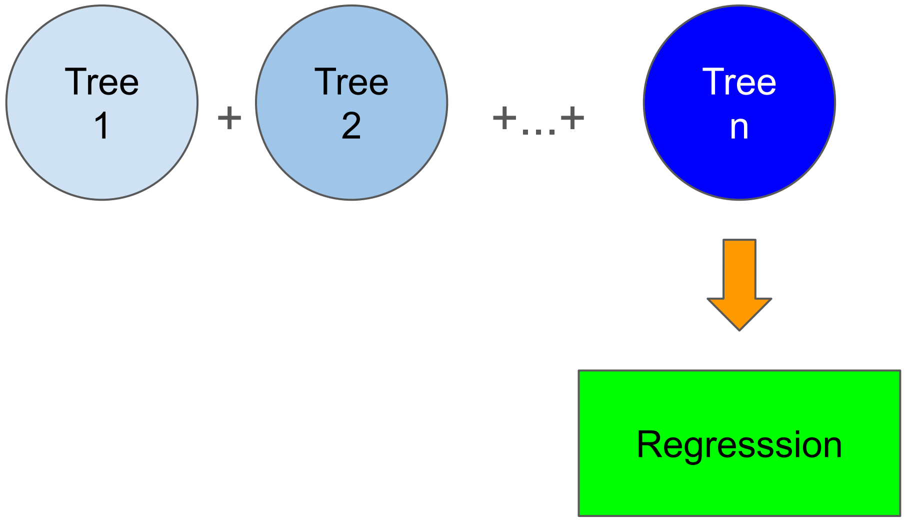
- **Good** at learning **discrete data**
- Extremely **prior to noises** in the data
- **Not robust** against a big amount of data
- Is **not** a **time series** model



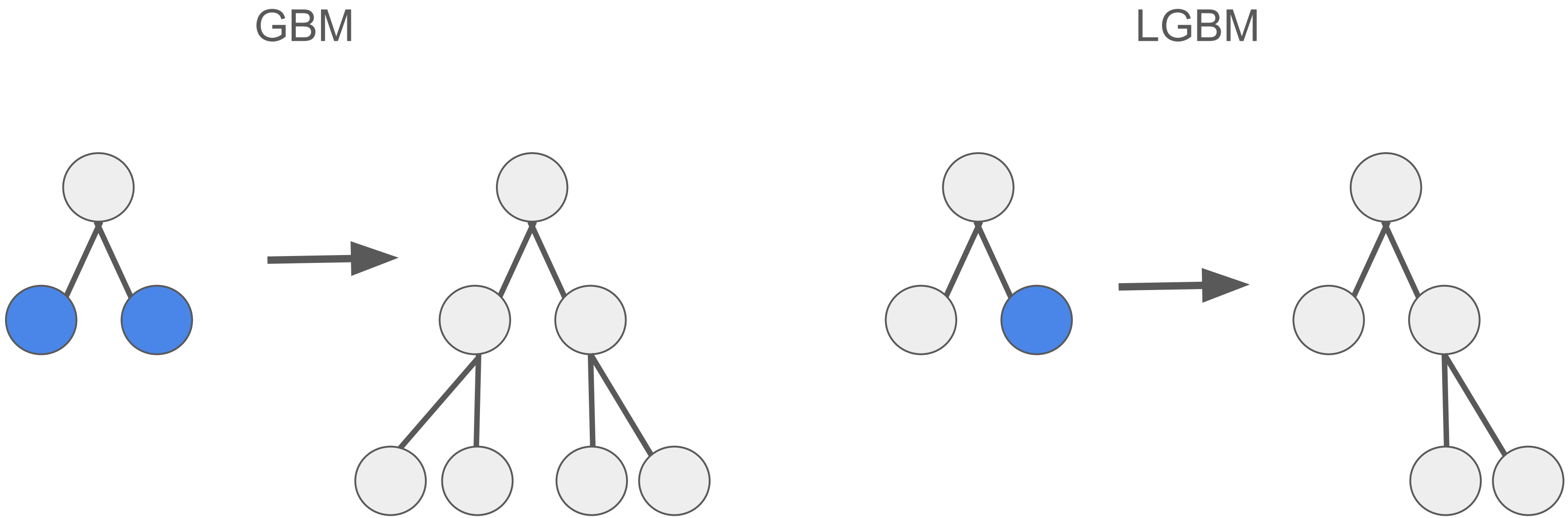
Our Model: Light Gradient Boosting Machine Regressor

Core characteristic :

- **Good** at learning **discrete data**
- **Prior to noises** in the data
- Is **not** a **time series** model
- **Need hyperparameter tuning**



LGBM vs GBM

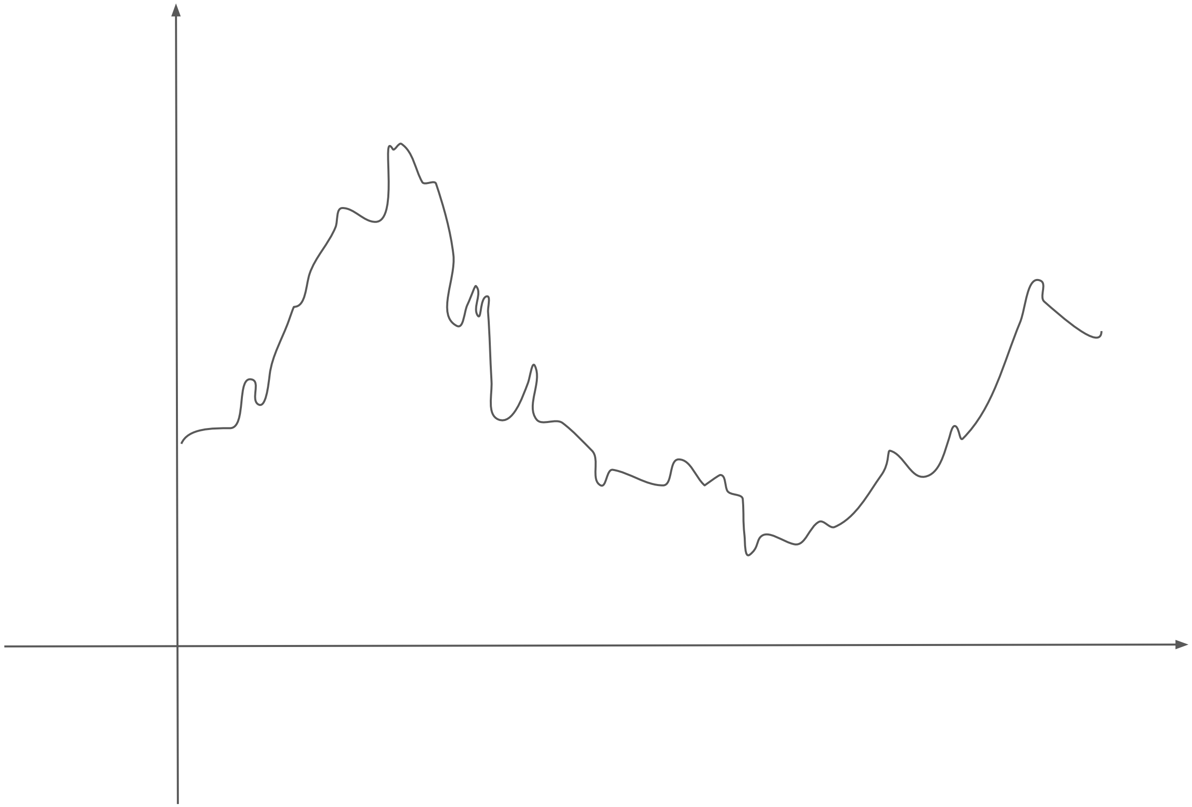


The Sliding Window

Sales:	Sales-1:	Sales-2:		Sales-n:
1	Nan	Nan		Nan
2	2	Nan		Nan
3	3	1	…	Nan
4	4	2		Nan
5	5	3		…
6	6	4		1

Before

The Sliding Window



Before

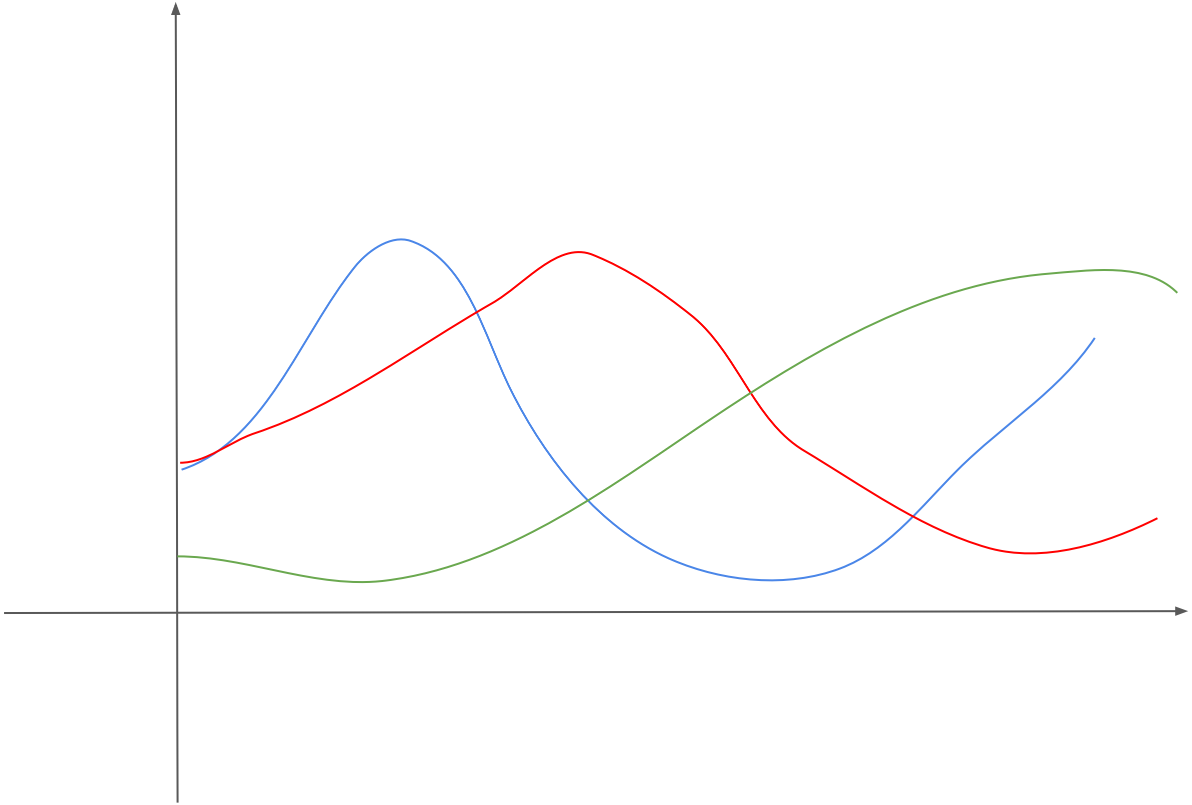
The Sliding Window

After

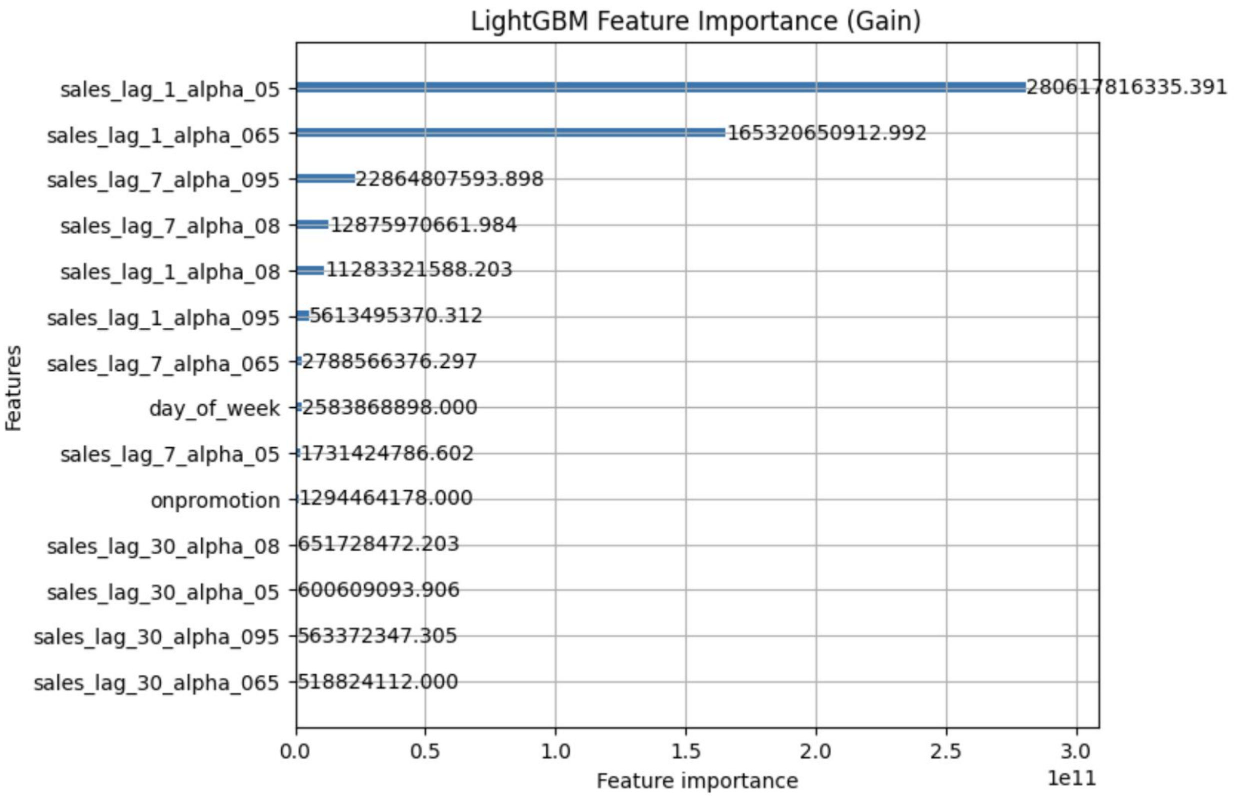
Sales:	Sales-1:	Sales-2:		Sales-n:
1	Nan	Nan		Nan
2	EMA(2)	Nan		Nan
3	EMA(3)	EMA(1)	…	Nan
4	EMA(4)	EMA(2)		Nan
5	EMA(5)	EMA(3)		…
6	EMA(6)	EMA(4)		EMA(1)

The Sliding Window

After



Gain of Each Features Ranked



Hyperparameter Tuning

Learning rate	: 0.1
Feature fracition	: 0.800087645
Bagging fraction	: 0.851134158
Bagging frequency	: 5
Verbose	: 0
Max depth	: 50
Num leaf	: 128
Max bin	: 512

What did we improve from before, and why?

- We **changed** our **model** from the regular **GBM** into **LGBM**.
- We **improve** our **sliding window**, which now gives better result
- We did **hyperparameter tuning**, the most important thing to do to **get better result** at GBM models.
- We **optimized** the **timeframe** used for training

Results

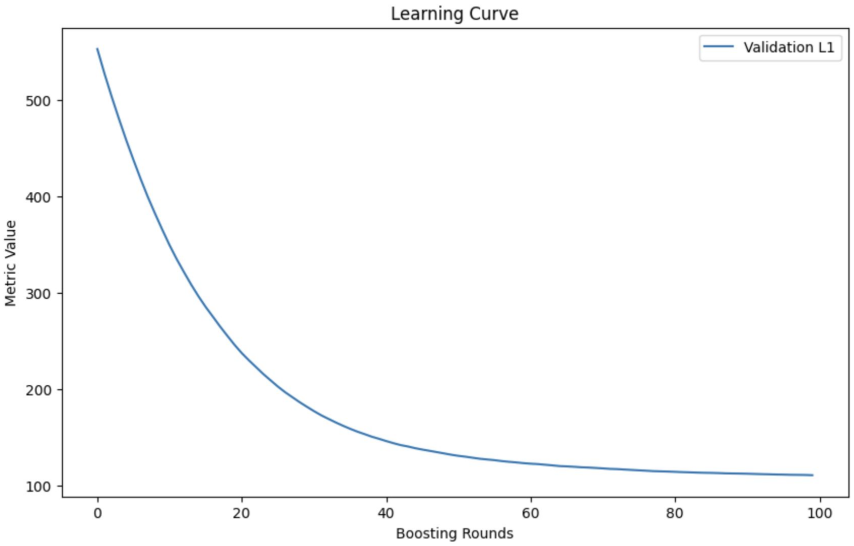
Our best result:

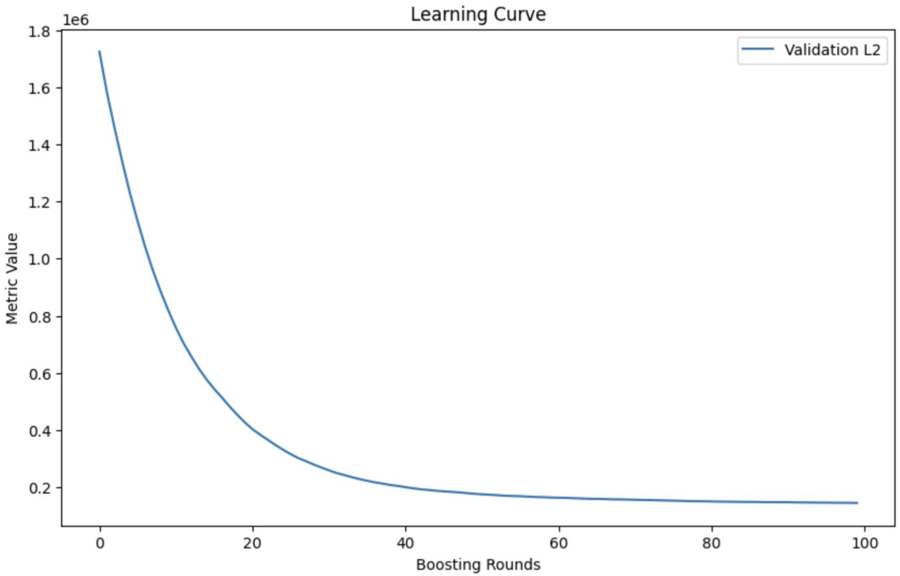
submission.csv

Complete · Ikhsan Rabbani · 11h ago · LGBM only train data without hp tuning

0.61289

Model Evaluation: Learning Curve





What did we learn?

Things We Notice….

- **Preprocessing** the data and exploratory data **analysis** are the **most important** thing to do before building a model

- **Sliding windows** can’t just be added, but also needs to be **optimized** to give best prediction for time series data

- **Hyperparameter** needs to be **fine tuned** to give the best result

Thank You

Any Question?