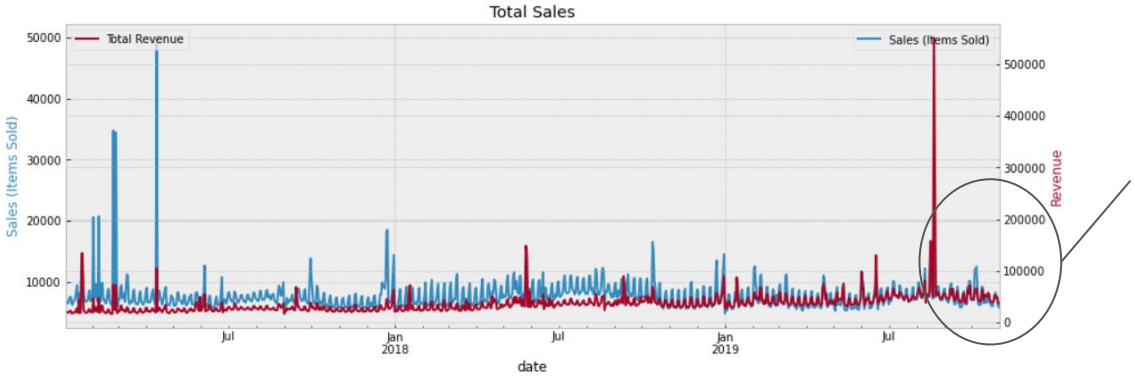
Task overview

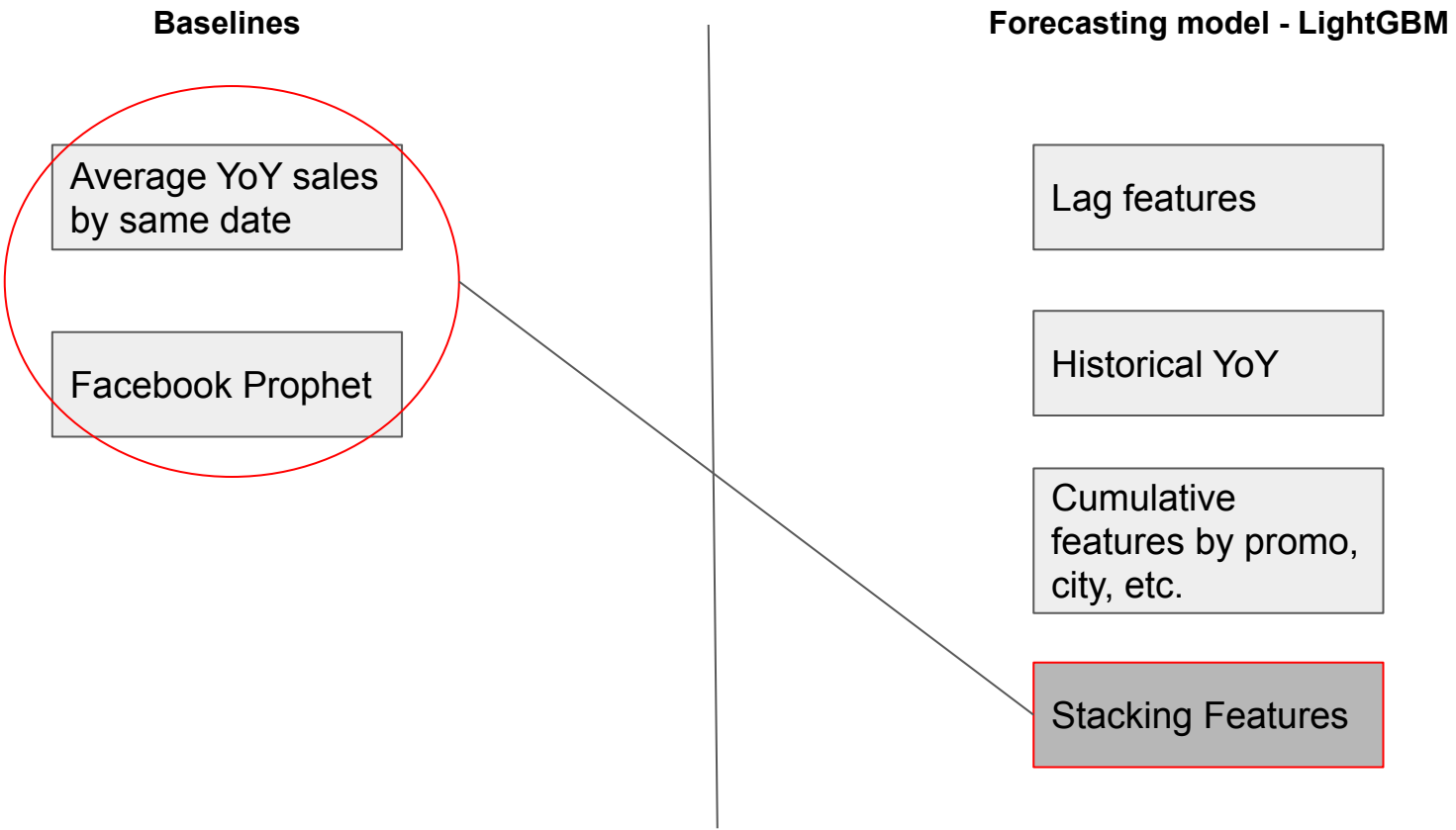


Sales forecast should be done for 496 data points

- 16 groups (4 hierarchy1_id by 4 storetype_id)
- 31 dates

[Exploratory data analysis]
[ML Validation]
[Github repo]

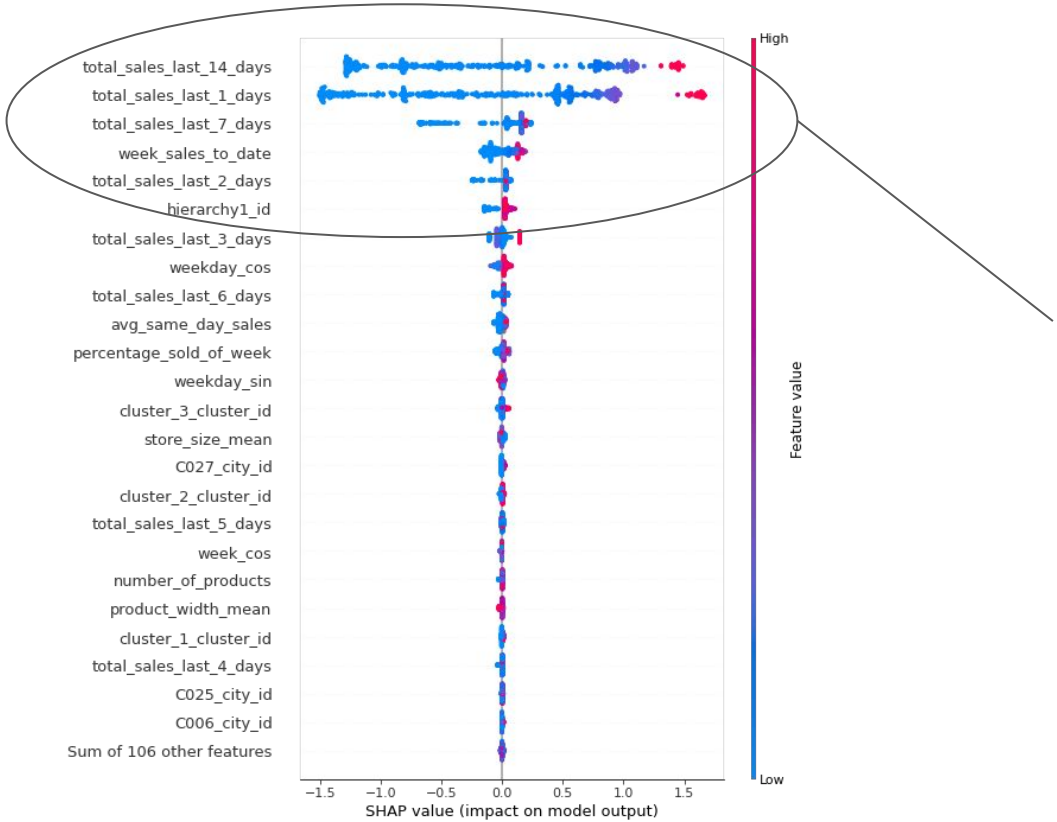
Model and features



Forecasting model - LightGBM with recursive predictions

With recursive predictions we are using previous predicted values as input features for new predictions

LightGBM - Feature Importance

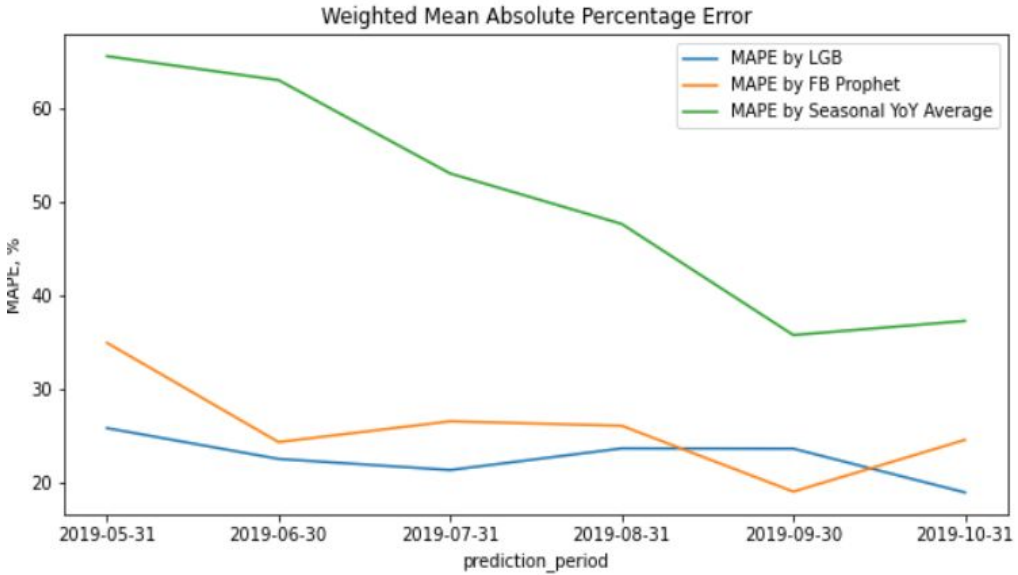


- 90% of model performance is based on 6 features
- Most important features are prior sales with 1, .., 14 days lag and YoY sales

Model performance - 05/2019 - 10/2019 (shown only groups with > 1% of sales)

Model performance is analyzed based on Weighted Mean Absolute Percentage Error

$$WMAPE = \frac{\sum_{t=1}^{n}(w_t|A_t - F_t|)}{\sum_{t=1}^{n}(w_t|A_t|)}$$



hierarchy1_id	storetype_id	WMAPE by LGB	WMAPE by FB Prophet	WMAPE by Seasonal YoY Average
H00	ST01	20%	21%	43%
	ST02	14%	14%	49%
	ST03	14%	15%	49%
	ST04	11%	12%	46%
H01	ST01	25%	30%	46%
	ST03	36%	39%	62%
	ST04	26%	28%	48%
H02	ST04	65%	74%	55%
H03	ST01	25%	27%	53%
	ST03	37%	46%	58%
	ST04	25%	33%	54%

Conclusion and Next steps

- Current approach showed significantly better performance than baselines but it gives a huge daily forecasting error
- There is a big disproportion between different product groups in terms of sales. It makes sense to apply different metrics and forecasting requirements to different products
- Additional features didn’t boost LightGBM. It makes sense to try Deep Learning approaches and/or more feature engineering for LightGBM

Next steps:

- Think about weighted error metrics for different product groups
- Think about weekly / monthly predictions instead of daily forecasting
- Add stock availability constraints to the model
- Feature engineering
- Reduce forecasting window
- Experiments with deep learning