M5 Forecasting - Accuracy
Estimate Walmart retail goods

LAI Cong (20747850)
LIU Jinghui (20745644)
LU Qiaoyu (20736916)
ZHEN Mengnan (20749066)

Contents

1.Introduction

- Forecast daily sales for the next 28 days
- Time series models
- Boosting models
- Deep learning models

2.Data description

In this project, we will **anticipate daily sales for the next 28 days** using hierarchical sales data from Walmart. The data comprises **item level, department, product categories, and store details** for stores in three US states. **Price, promotions, day of the week, and special events** are among the explanatory variables.

- Calendar - information about the dates on which the products are sold.
- sales_train_validation - the historical daily unit sales data per product and store [d_1 - d_1913]
- sample_submission - The correct format for submissions.
- sell_prices - information about the price of the products sold per store and date.
- sales_train_evaluation - sales [d_1 - d_1941]

Time series models

- Data preprocess: SVD
- Model details: EMA, SARIMA, STL

TS
- Data preprocess

id	item_id	dept_id	cat_id	store_id	state_id	d_1	d_2	d_3	d_4	d_5	...	d_1932	d_1933	d_1934	d_1935	d
HOBBIES_1_001_CA_1_evaluation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	0	...	2	4	0	0	
HOBBIES_1_002_CA_1_evaluation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	0	...	0	1	2	1	
HOBBIES_1_003_CA_1_evaluation	HOBBIES_1_003	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	0	...	1	0	2	0	
HOBBIES_1_004_CA_1_evaluation	HOBBIES_1_004	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	0	...	1	1	0	4	
HOBBIES_1_005_CA_1_evaluation	HOBBIES_1_005	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	0	...	0	0	0	2	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
FOODS_3_823_WI_3_evaluation	FOODS_3_823	FOODS_3	FOODS	WI_3	WI	0	0	2	2	0	...	1	0	3	0	
FOODS_3_824_WI_3_evaluation	FOODS_3_824	FOODS_3	FOODS	WI_3	WI	0	0	0	0	0	...	0	0	0	0	
FOODS_3_825_WI_3_evaluation	FOODS_3_825	FOODS_3	FOODS	WI_3	WI	0	6	0	2	2	...	0	0	1	2	
FOODS_3_826_WI_3_evaluation	FOODS_3_826	FOODS_3	FOODS	WI_3	WI	0	0	0	0	0	...	1	1	1	4	
FOODS_3_827_WI_3_evaluation	FOODS_3_827	FOODS_3	FOODS	WI_3	WI	0	0	0	0	0	...	1	2	0	5	

store_id	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_10	...	d_1932	d_1933	d_1934	d_1935	d_1936	d_1937	d_1938	d_1939	d_1940	d_1941
CA_1	0	0	0	0	0	0	0	0	0	0	...	2	4	0	0	0	0	3	3	0	1
CA_2	0	0	0	0	0	0	0	0	0	0	...	2	0	2	0	2	2	0	2	0	1
CA_3	0	0	0	0	0	0	0	0	0	0	...	2	6	0	1	0	2	1	0	1	0
CA_4	0	0	0	0	0	0	0	0	0	0	...	1	0	3	1	1	1	0	1	2	2
TX_1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	2	1	0	2	1	0	1
TX_2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	2	0	0	0	1
TX_3	0	0	0	0	0	0	0	0	0	0	...	1	0	3	0	0	3	1	1	2	1
WI_1	0	0	0	0	0	0	0	0	0	0	...	0	1	0	2	0	0	0	0	1	2
WI_2	0	0	0	0	0	0	0	0	0	0	...	0	1	0	0	0	0	0	0	0	0
WI_3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

Transform original pattern

The evaluation and the validation data contain many zeros. To iterated over the departments, producing a data matrix of unit sales of each product in all 10 stores during the time range of training dataset.

SVD

Reduce noise and obtain important parts from data matrix

Product data

First, performed SVD and then replacing the data with a reduced-rank approximation of itself, where we selected top 90% of the singular values of each product.

TS
- Model details

EMA

- Smoothing time series data using the exponential window function
- Assign exponentially decreasing weights over time
- Seasonality

SARIMA

- Low order of AR and MA
- Seasonal / nonseasonal

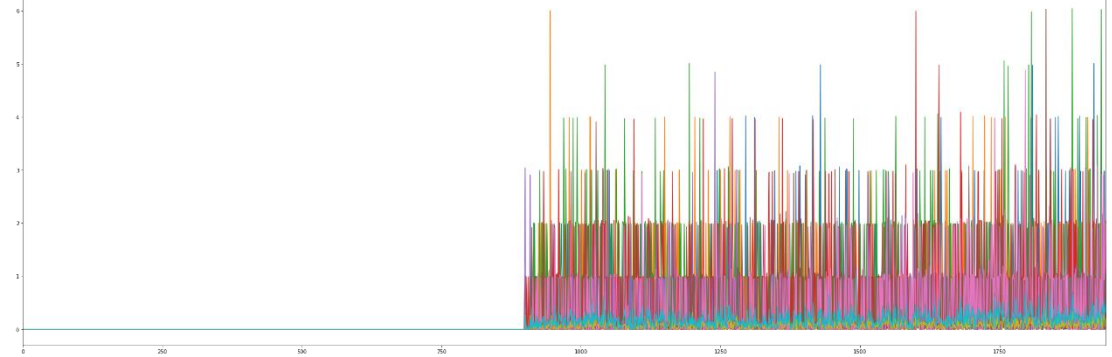
STL

- Seasonal and Trend decomposition using Loess
- Trend: general direction of the overall data
- Seasonality: regular and predictable pattern that recur at a fixed interval of time
- Loess: regression technique that uses local weighted regression to fit time series

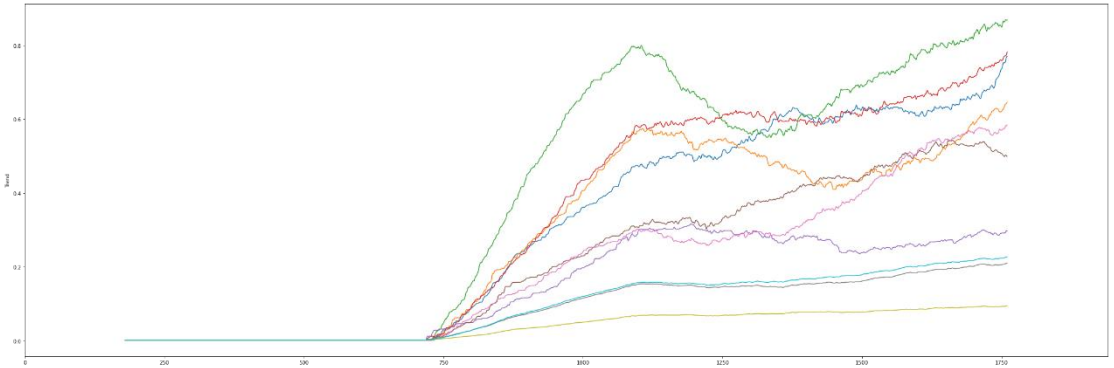
• STL

- Product unit sales of 10 stores in the time interval of training dataset

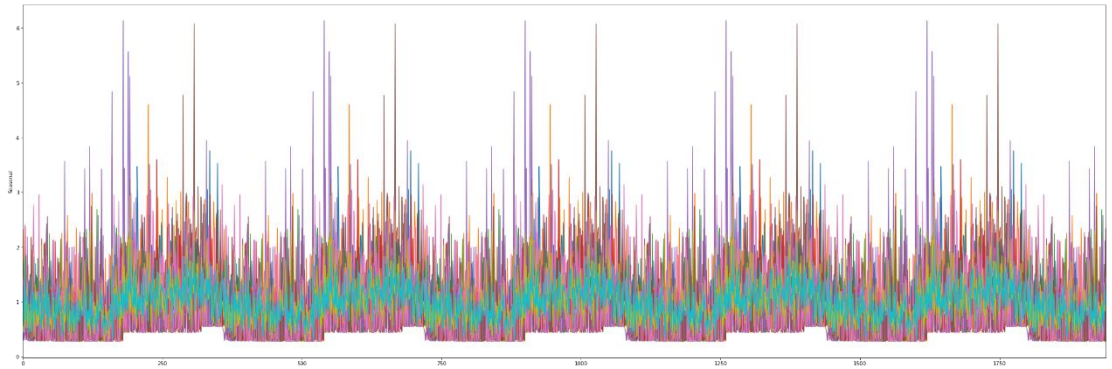
- Time series



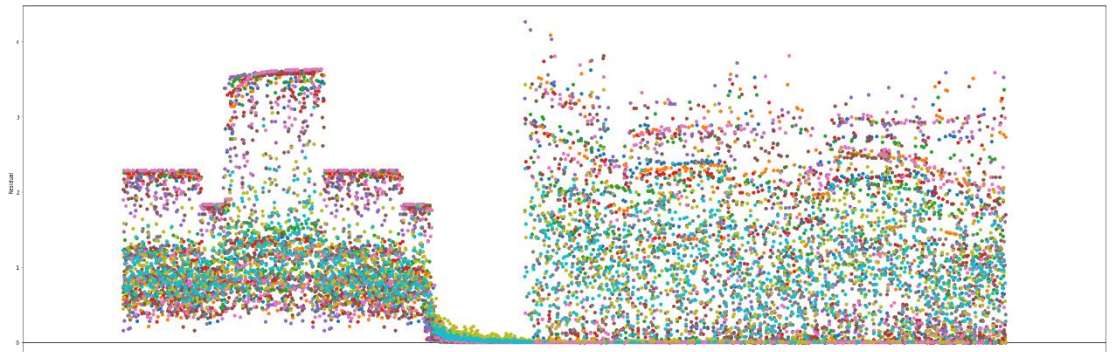
- Trend



- Seasonality



- Residual



TS
- Model selection

Using $RMSE$ to find optimal components.

Then chose the parameter which had the minimum $RMSE$ every time.

Parameters	Description	Values
Order	Order for ARIMA, EMA	(0,1,0) to (1,1,1)
Seasonal	Seasonal term or seasonal differencing	365 or none

For simplicity, we only show the results of models with optimal order parameters. And because of poor STL results of predicting revenues we stopped further studying of the others models on it.

From the table above, we can see EMA, SARIMA models performed extremely bad.

Consequently, we select STL model with seasonal = 365 and ARIMA (1,1,1) as forecasting model as the best model among time series models.

Models	Seasonal	RMSE (Train)	RMSE (Validation)	Kaggle score
EMA	Yes	445006756395	4096894337805	13350202691217
EMA	None	466006756395	4796894337805	
ARIMA	None	405006755397	4099884437826	
SARIMA	Yes	366011756395	3796894337805	
STL	Yes	2.330	2.429	0.780
STL	None	2.920	3.399	0.845

Extreme Gradient Boosting (XGBoost)

- Modeling process
- Model details
- Model selection

XGBoost
- Modeling process

Data splitting:

- Train dataset (‘date’ before ‘2016-03-27’),
- Validation dataset (‘date’ between ‘2016-03-27’ and ‘2016-04-25’)
- Test dataset (‘date’ after ‘2016-04-25’)

Feature Engineering:

- Dummy variable conversion
- Data type conversion to save memory usage
- Feature construction based on the original dataset

rolling_mean_t365	rolling_std_t30	price_change_t1	price_change_t365
0.6600	0.9277	0.0	0.0
0.3562	0.4795	0.0	0.0
0.5015	1.1670	0.0	0.0
2.0140	2.0760	0.0	0.0
1.1120	0.9595	0.0	0.0

XGBoost
- Model details

Model introduction:

Extreme Gradient Boosting (XGBoost) is an advanced machine learning model to utilize the computations resources deeply, which efficiently implements GBDT algorithm.

Strengths of XGBoost:

- Automatically apply the missing value handling strategy
- Support data sampling and various types of classifiers
- Utilize the first and second derivatives of the cost function
- Optimize model prediction by avoiding overfitting

XGBoost
- Model selection

Parameter selecting:

Using $RMSE$ to find optimal components.
Then chose the parameter which had the minimum validation $RMSE$.

Parameter	Description	Range
n_estimators	Number of weak classifiers in decision tree	[5, 10, 30, 50, 100]
learning_rate	Control the weight reduction coefficient of each weak classifier	[0.01, 0.05, 0.1, 0.2]

submission:

The best XGBoost model:
learning rate is 0.2 and number of estimators is 100

	n_estimators				
Learning_rate	5	10	30	50	100
0.01	1.18980	1.17648	1.13273	1.10176	1.05982
0.05	1.14121	1.10051	1.04454	1.03853	1.03640
0.1	1.09884	1.05771	1.03747	1.03616	1.03411
0.2	1.05659	1.04008	1.03630	1.03487	1.03316

The $RMSE$ of each model

Light Gradient Boosting Machine Model (LGBM)

- Modeling process
- Model details

LGBM
- Modeling process

Data Pre-processing:

The datasets used:

For train and validation : sales_train_evaluation (d_1-d_1941)

For test : add zero sales for dates d_1942 to d_1969

Melt & Merge datasets:

Melt: melt dataset to aggregate sales by ID

Merge: merge sales_train_evaluation, calendar and sell_prices

Function: memory usage reduction

save memory for datasets.

Variables conversion:

converted numeric variables into categorical variables for model-building.

LGBM
- Model details

Model introduction:

LightGBM is a framework for implementing the GBDT method, which provides efficient parallel training and has the benefits of faster training speed, lower memory usage, improved accuracy, distributed support, and rapid data processing.

XGBoost VS. LightGBM :

XGBT	LGBM
• space consumption • a large cost in time	• Decision tree algorithm based on Histogram. • Gradient-based One-side Sampling (GOSS) : Save space • Exclusive Feature Bundling (EFB)

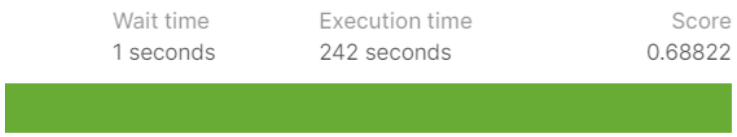
LGBM
- Model details

Parameter selecting:

Using $RMSE$ to find optimal components.
Then chose the parameter which had the minimum $RMSE$ every time.

Parameter	Description	Range
max_depth	Maximum depth of each tree to prevent overfitting.	[3,4,6]

submission:



Chose max_depth=6, which had the minimum $RMSE$ of validation dataset, for further model prediction.

max_depth	Category	RMSE(train)	RMSE(validation)	
3	HOBBIES	1.70545	1.69749	
	HOUSEHOLD	1.64616	1.78818	
	FOODS	4.10085	3.83612	
		7.45246	7.32179	SUM
4	HOBBIES	1.68301	1.67881	
	HOUSEHOLD	1.54367	1.66867	
	FOODS	3.77333	3.50963	
		7.00001	6.85711	SUM
6	HOBBIES	1.65119	1.6589	
	HOUSEHOLD	1.41275	1.53633	
	FOODS	3.309	3.10543	
		6.37294	6.30066	SUM

The $RMSE$ of each model

Neural Networks (NN)

- A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.

NN
- Modeling process

Dataset for train and validation

sales_train_evaluation dataset (from d_1 to d_1941 or from d_1602 to d_1941)

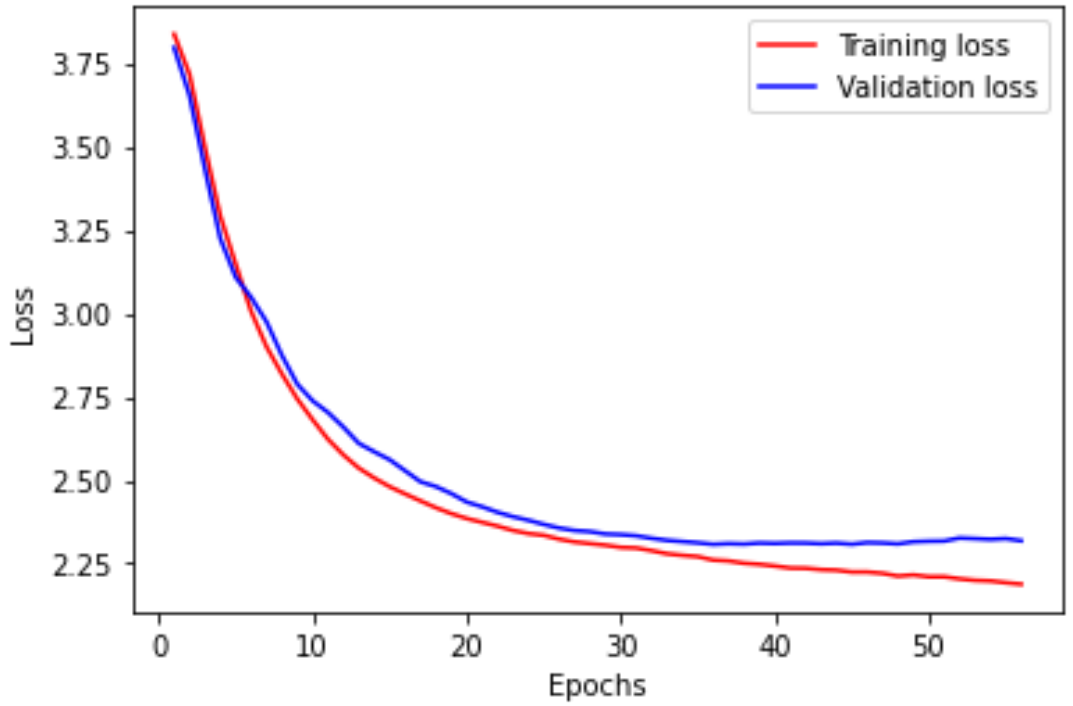
Variables conversion

process the events columns to binary (any event: 1, no event: 0)

NN
- Model details

Model construction

- Type: simple NN model with 2 hidden layers
- Active function: ReLU
- Optimizer: adam
- Loss function: RMSE



NN
- Performance

First, we use whole dataset (d_1 – d_1941) and extract 75% sample data randomly to train the model. Then we reduce the sample size and train model again.

When we train our NN model with smaller dataset, the model has better performance.

	75%	66.7%
d_1 – d_1941	2.09361	2.05081
d_1602 – d_1941	0.96589	0.93854

Comparison

Model	Performance (RMSE)
LGBM	0.68822
TS	0.78559
XGBoost	0.86330
NN	0.93854

Thanks!