Predict Future Sales

Final Report

Kaicheng Jia

CONTENTS

01 INTRODUCTION

Background

• In this project we will work with a challenging timeseries dataset consisting of daily sales data, kindly
provided by one of the largest Russian software firms - 1C Company.

• The task is to forecast the total amount of products sold in every shop for the test set in the next month.

02 DATA

Files

File descriptions

- sales_train.csv - the training set. Daily historical data from January 2013 to October 2015.
- test.csv - the test set. You need to forecast the sales for these shops and products for November 2015.
- sample_submission.csv - a sample submission file in the correct format.
- items.csv - supplemental information about the items/products.
- item_categories.csv - supplemental information about the items categories.
- shops.csv- supplemental information about the shops.

Basic info of train

```
sales_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2935849 entries, 0 to 2935848
Data columns (total 6 columns):
date            object
date_block_num  int64
shop_id         int64
item_id         int64
item_price      float64
item_cnt_day    float64
dtypes: float64(2), int64(3), object(1)
memory usage: 134.4+ MB
```

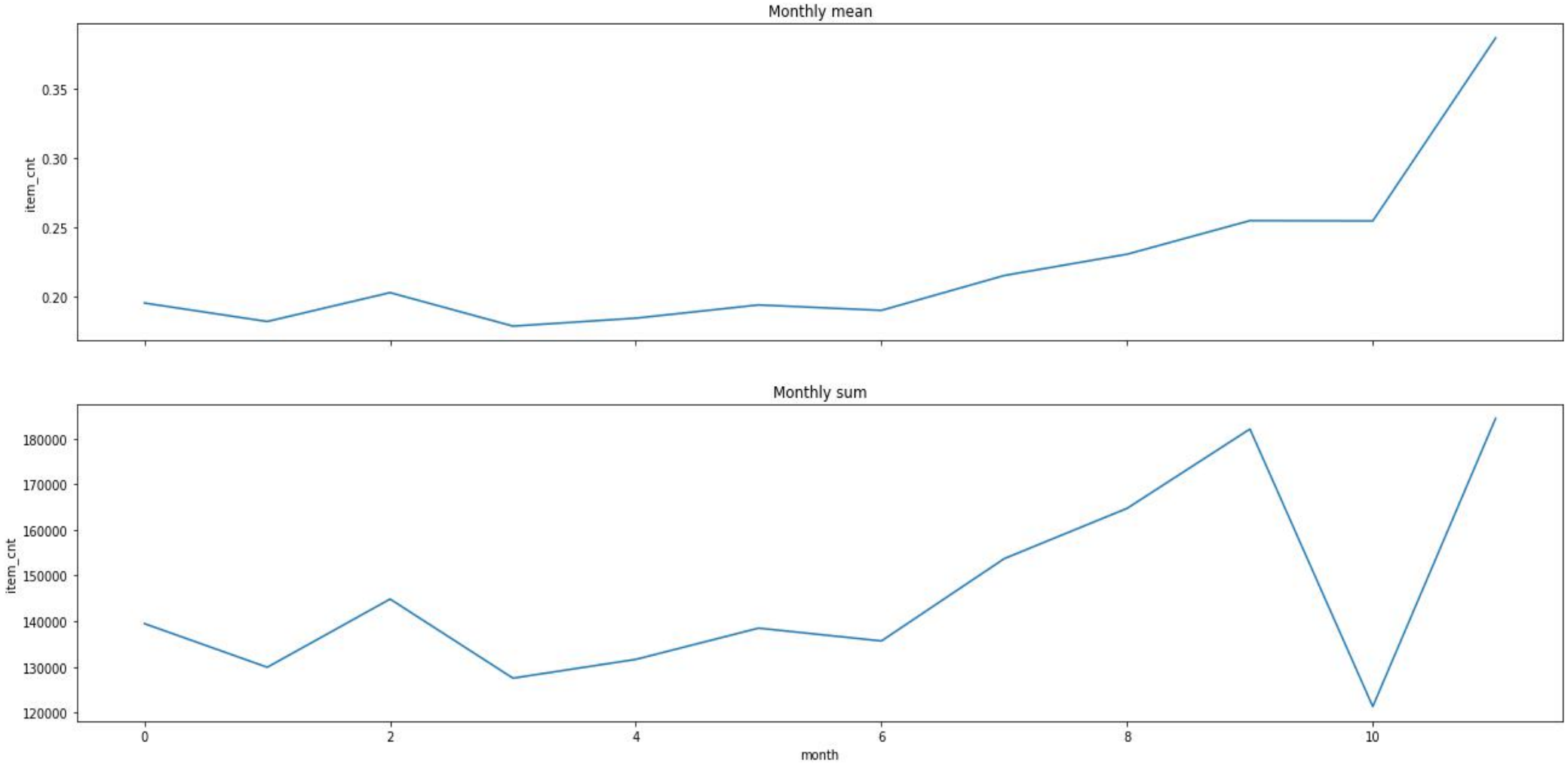
	date	date_block_num	shop_id	item_id	item_price	item_cnt_day
0	02.01.2013	0	59	22154	999.00	1.0
1	03.01.2013	0	25	2552	899.00	1.0
2	05.01.2013	0	25	2552	899.00	-1.0
3	06.01.2013	0	25	2554	1709.05	1.0
4	15.01.2013	0	25	2555	1099.00	1.0

date - date in format dd/mm/yyyy
date_block_num - a consecutive month number, used for convenience. January 2013 is 0
shop_id - unique identifier of a shop
item_id - unique identifier of a product
item_price - current price of an item
item_cnt_day - number of products sold. predicting a monthly amount of this measure

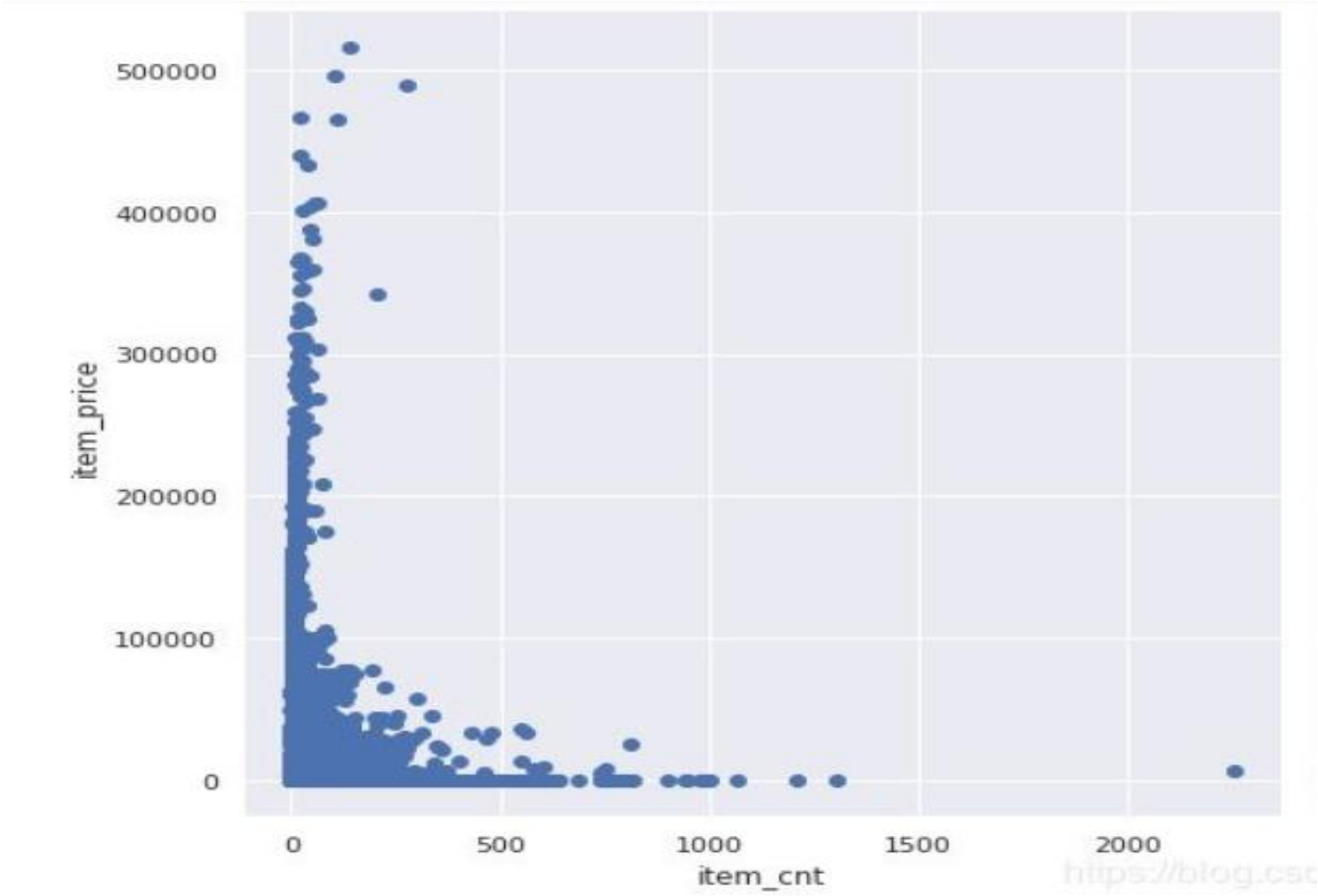
Duplicated
And Missing values

```
duplicated lines in sales_train is 5
missing value in sales_train is
date                  0
date_block_num        0
shop_id               0
item_category_id      0
item_id               0
item_price            0
item_cnt_day          0
dtype: int64
```

Item_cnt month trend



Item_price VS item_cnt



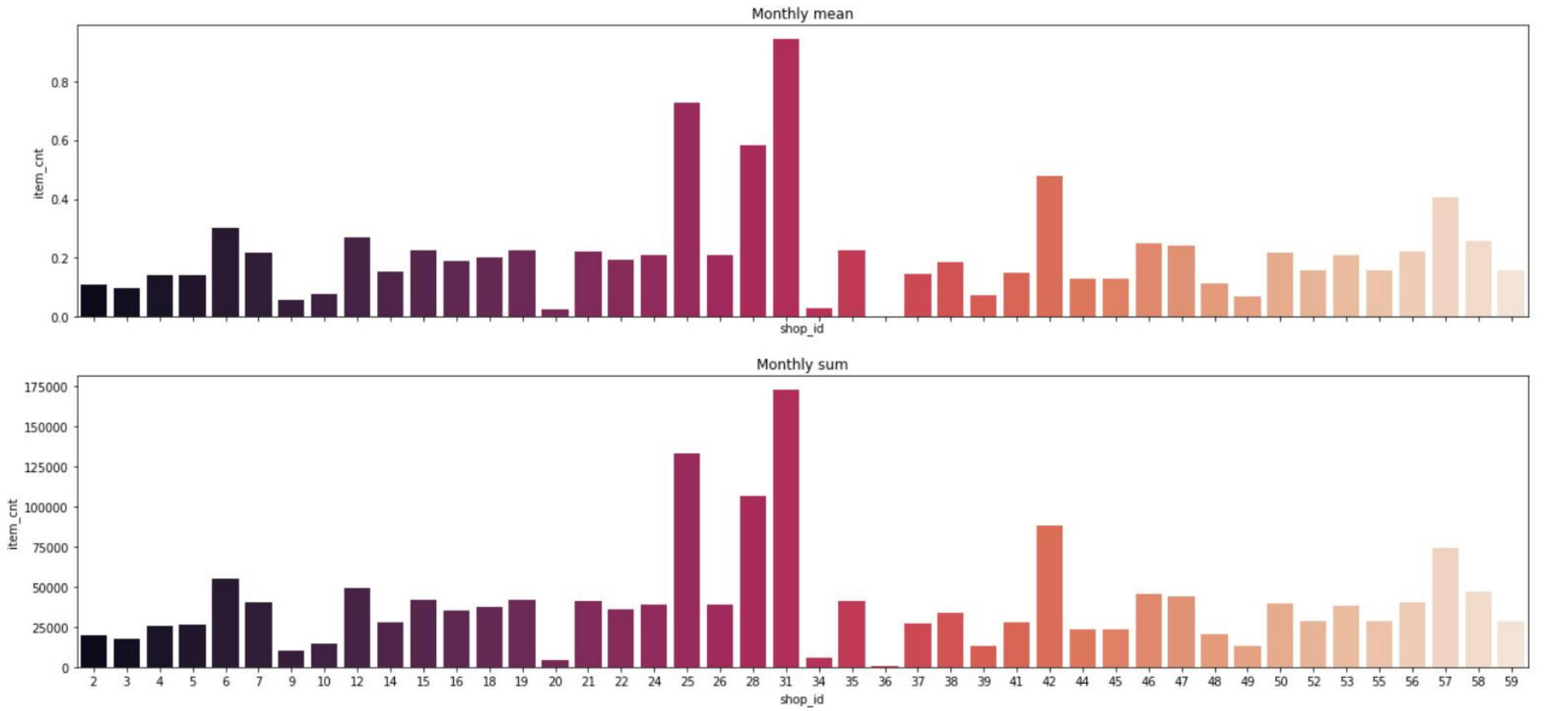
Basic info of shops

	shop_name	shop_id
0	!Якутск Орджоникидзе, 56 фран	0
1	!Якутск ТЦ "Центральный" фран	1
2	Адыгея ТЦ "Мега"	2
3	Балашиха ТРК "Октябрь-Киномир"	3
4	Волжский ТЦ "Волга Молл"	4

Shop_name: City|type|name

Волжский ТЦ "Волга Молл"	Volga shopping center "Volga Mall"

Shops VS item_cnt



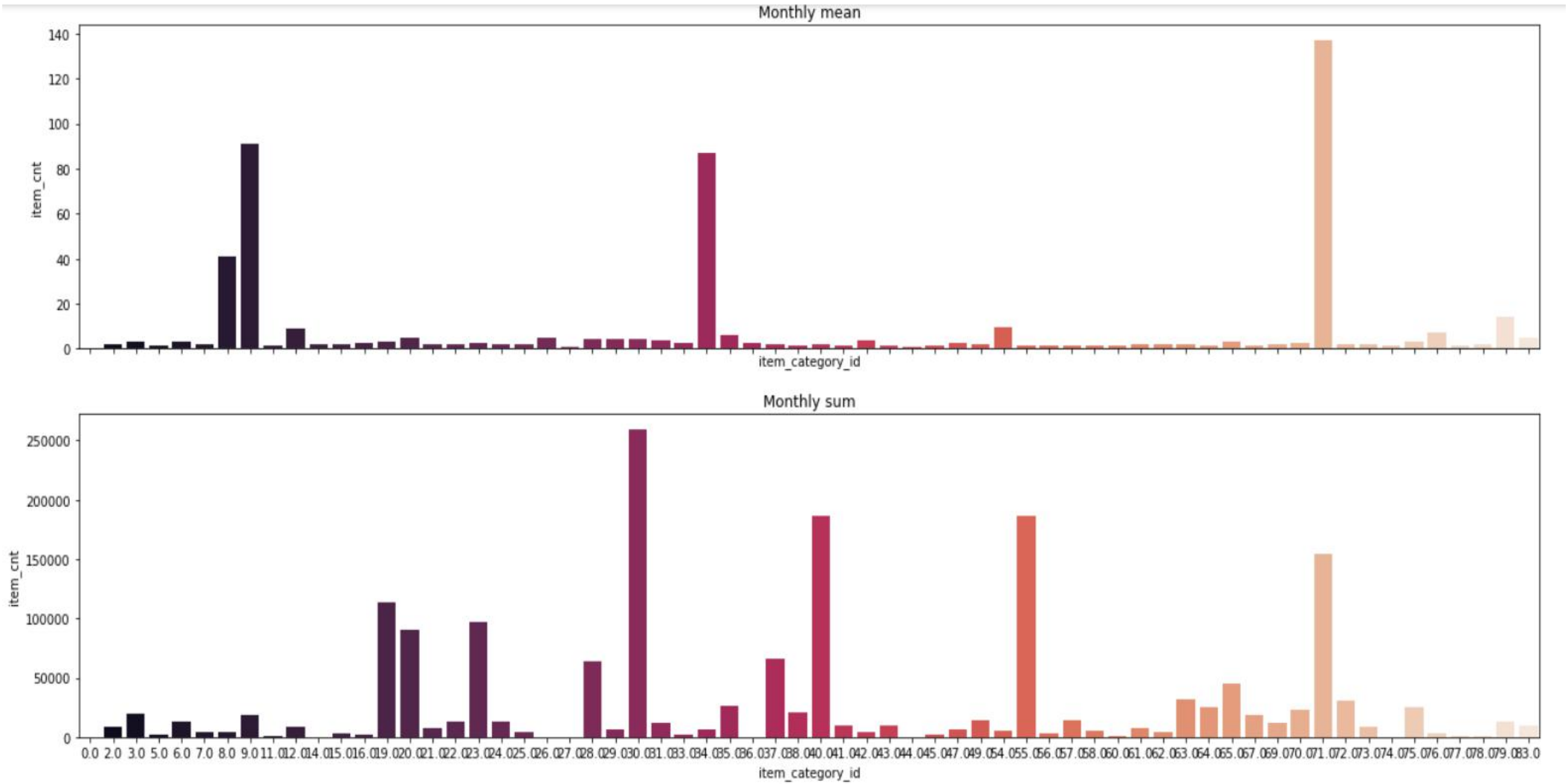
Basic info of item_category

	item_category_name	item_category_id
0	PC - Гарнитуры/Наушники	0
1	Аксессуары - PS2	1
2	Аксессуары - PS3	2
3	Аксессуары - PS4	3
4	Аксессуары - PSP	4

Item_category_name: category|sub_cat

Служебные - Билеты	Service - Tickets
Чистые носители (шпиль)	Net carriers (spire)

item_category VS item_cnt



Basic info of items

	item_name	item_id	item_category_id
0	! ВО ВЛАСТИ НАВАЖДЕНИЯ (ПЛАСТ.) D	0	40
1	!ABBYY FineReader 12 Professional Edition Full...	1	76
2	***В ЛУЧАХ СЛАВЫ (UNV) D	2	40
3	***ГОЛУБАЯ ВОЛНА (Univ) D	3	40
4	***КОРОБКА (СТЕКЛО) D	4	40

Item_name: name|sub_type1|sub_type2

! ВО ВЛАСТИ НАВАЖДЕНИЯ (ПЛАСТ.) D	! VO VLASTI NAVAZHDENIYA (PLAST.) D

Basic info of test

	ID	shop_id	item_id
0	0	5	5037
1	1	5	5320
2	2	5	5233
3	3	5	5232
4	4	5	5268

```
how many lines in train set: (2935849, 6)
unique items in train set: 21807
unique shops in train set: 60
how many lines in test set: (214200, 3)
unique items in test set: 5100
unique shops in test set: 42
```

- ID - an Id that represents a (Shop, Item) tuple within the test set

Feature Engineering

- Data Preprocessing (won’t be mentioned here)
- Split features & Label Encoding
- Lag features & Mean Encoding
- Dummy variable & Other features
- Merge features

Split features & Label Encoding

item_cnt_day

item_price

sum → item_cnt_month

mean → item_price_month

item_name

shop_name

item_category_name

split name + Label Encoding →

item_f1_code

city_code

type_code

item_f2_code

shop_type_code

subtype_code

ID

item_id

shop_id

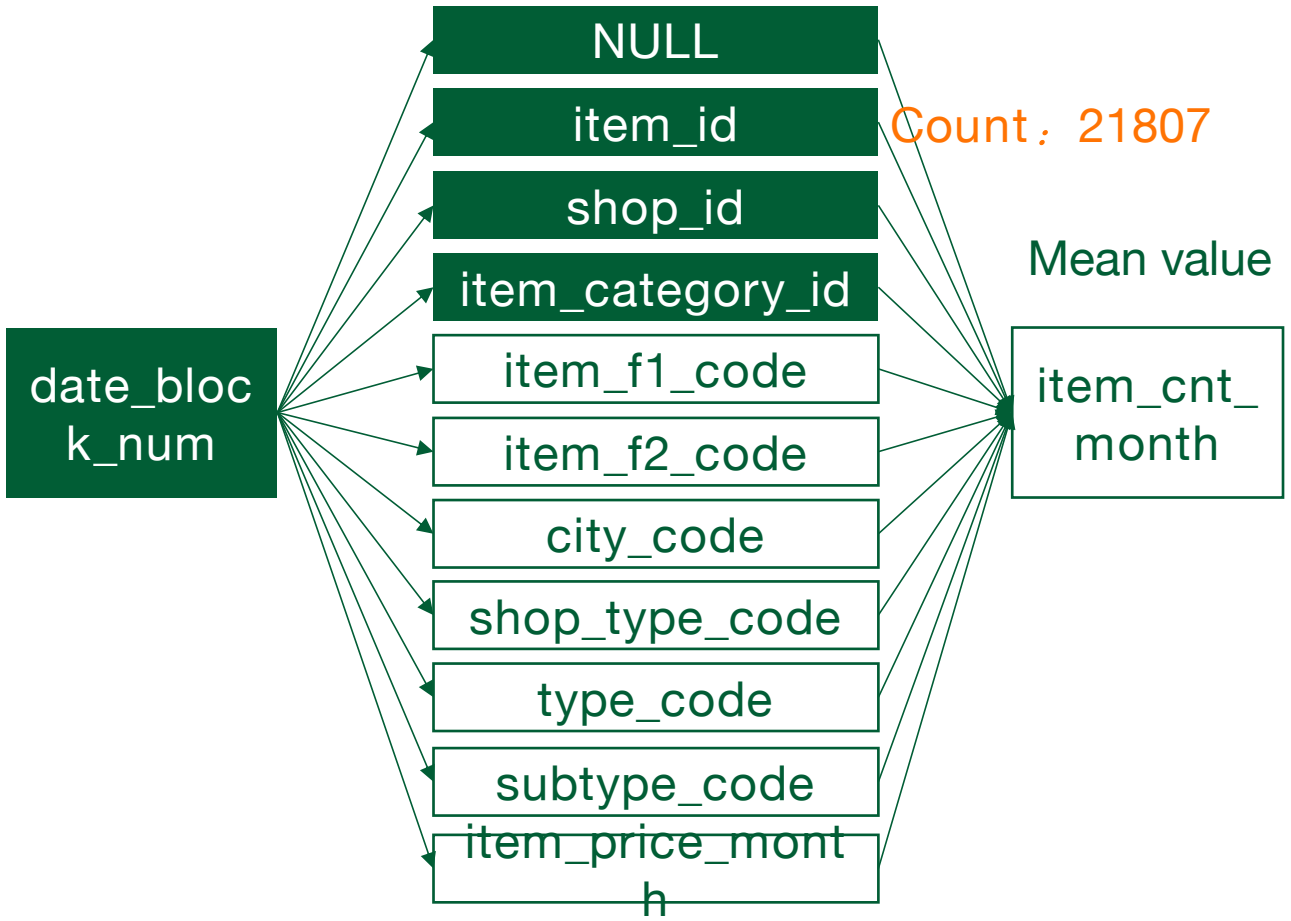
item_category_id

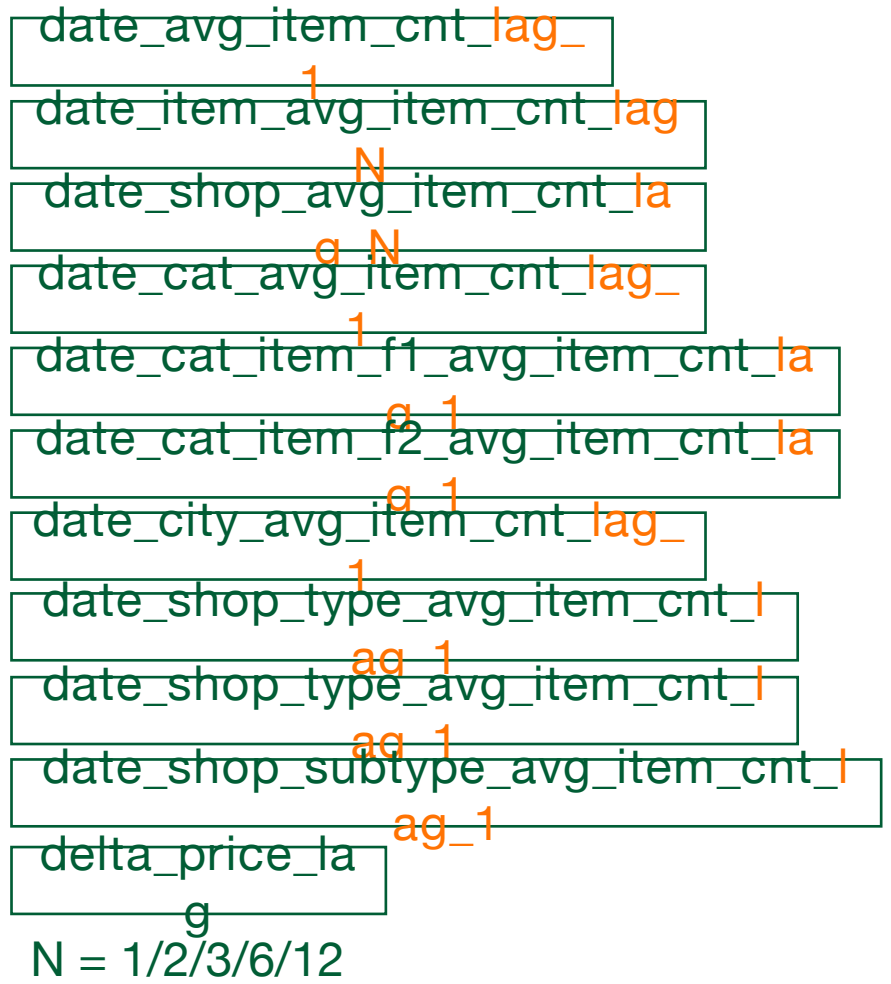
date

date_block_num

city_code	shop_type_code	item_category_id	item_f1_code	item_f2_code	type_code	subtype_code
0	5	40	729	138	10	4
0	5	19	729	138	5	10
0	5	30	729	138	8	53
0	5	23	729	138	5	16
0	5	40	729	138	10	4
...	...	...	...	...	...	...
20	5	55	729	138	12	2
20	5	64	729	138	13	41
20	5	55	729	138	12	2
20	5	40	853	138	10	4
20	5	37	364	138	10	1

Lag features & Mean Encoding





N = 1/2/3/6/12

- Mean Encoding:
  Sum “item_cnt_month” of the same “data_block_num” and the features in the center column, then divide its frequency.

Dummy variable & Other features

0 - 33	1 - 12	0 / 1
date_block_num	month	dummy

- From Jan to Nov, set dummy value to 0
- In Dec, set dummy value to 1

hist_min_item_price	The lowest daily price per item in history
hist_max_item_price	The highest daily price per item in history
item_first_sale	The first daily sales of an item in history
item_shop_first_sale	The first daily sales of a pair in history
holidays	The number of Russian official holidays in every month

Merge features

- Original features: 4
- Split features & Label Encoding: 8
- Lag features & Mean Encoding: 19
- Dummy variable & Other features: 7

	date_block_num	item_cnt_month	item_id	shop_id	city_code	shop_type_code	item_category_id	item_f1_code
0	0	0.0	19	2	0	5	40	729
1	0	1.0	27	2	0	5	19	729
2	0	0.0	28	2	0	5	30	729
3	0	0.0	29	2	0	5	23	729
4	0	0.0	32	2	0	5	40	729
...	...	...	...	...	...	...	...	...
11056272	34	0.0	18454	45	20	5	55	729
11056273	34	0.0	16188	45	20	5	64	729
11056274	34	0.0	15757	45	20	5	55	729
11056275	34	0.0	19648	45	20	5	40	853
11056276	34	0.0	969	45	20	5	37	364

11056276 rows x 39 columns

03 ALGORITHM

ALGORITHMs

LinearRegression

A linear approach to modeling the relationship between a scalar response and one or more explanatory variables. The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression.

Not employed

RandomForest

An ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction (regression) of the individual trees.

employed

XGBoost

XGBoost implements machine learning algorithms under the Gradient Boosting framework. It provides a parallel tree boosting that solve many data science problems in a fast and accurate way.

employed

LightGBM

A fast, distributed, high performance gradient boosting (GBT, GBDT, GBRT, GBM or MART) framework based on decision tree algorithms, used for ranking, classification and many other machine learning tasks

employed

Prediction without Parameters Tuning

EVALUATION METRIC :

Root Mean Square Error均方根误差

$$RMSE(X,h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2}$$

	Train_rmse	Valid_rmse	Test_rmse
lightGBM	0.69266	0.89392	0.91685
XGBoost	0.68734	0.89816	0.91812

Parameters Tuning

XGBoost Parameters:

The overall parameters have been divided into 3 categories by XGBoost authors:

1. General Parameters: Guide the overall functioning
2. Booster Parameters: Guide the individual booster (tree/regression) at each step
3. Learning Task Parameters: Guide the optimization performed

Parameters Tuning

8 Booster Parameters:

eta (default=0.3, alias: learning_rate)
Step size shrinkage used in update to prevents overfitting. After each boosting step, we can directly get the weights of new features, and eta shrinks the feature weights to make the boosting process more conservative.
range: [0,1]

gamma (default=0, alias: min_split_loss)
Minimum loss reduction required to make a further partition on a leaf node of the tree. The larger gamma is, the more conservative the algorithm will be.
range: [0,∞)

max_depth (default=6)
Maximum depth of a tree. Increasing this value will make the model more complex and more likely to overfit. 0 is only accepted in lossguided growing policy when tree_method is set as hist and it indicates no limit on depth. Beware that XGBoost aggressively consumes memory when training a deep tree.
range: [0,∞)

min_child_weight (default=1)
Minimum sum of instance weight needed in a child. If the tree partition step results in a leaf node with the sum of instance weight less than min_child_weight, then the building process will give up further partitioning. In linear regression task, this simply corresponds to minimum number of instances needed to be in each node. The larger min_child_weight is, the more conservative the algorithm will be.
range: [0,∞)

Parameters Tuning

8 Booster Parameters:

subsample (default=1)
Subsample ratio of the training instances. Setting it to 0.5 means that XGBoost would randomly sample half of the training data prior to growing trees. and this will prevent overfitting. Subsampling will occur once in every boosting iteration.
range: (0,1]
colsample_bytree (default=1)
The subsample ratio of columns when constructing each tree. Subsampling occurs once for every tree constructed.
range: (0, 1], the default value of 1, and specify the fraction of columns to be subsampled.

alpha (default=0, alias: reg_alpha)
L1 regularization term on weights. Increasing this value will make model more conservative.
lambda (default=1, alias: reg_lambda)
L2 regularization term on weights. Increasing this value will make model more conservative.

1 other Parameters:

n_estimators (alias: num_boosting_rounds)

Parameters Tuning

Tuning Methods:

- Early Stopping

```
model = xgb.train(params, d_train, 10000, watchlist, early_stopping_rounds=100, verbose_eval=10)
```

Train until valid-rmse hasn't improved in 100 rounds.

- sklearn.model_selection.GridSearchCV from Scikit-learn

```
CV :
cv_params = {'n_estimators': [550, 575, 600, 650, 675]}
        other_params = {'learning_rate': 0.1,  'max_depth': 5, 'min_child_weight': 1,
                                'colsample_bytree': 0.8, 'gamma': 0,
                        'reg_alpha': 0, 'reg_lambda': 1}
```

Parameters Tuning

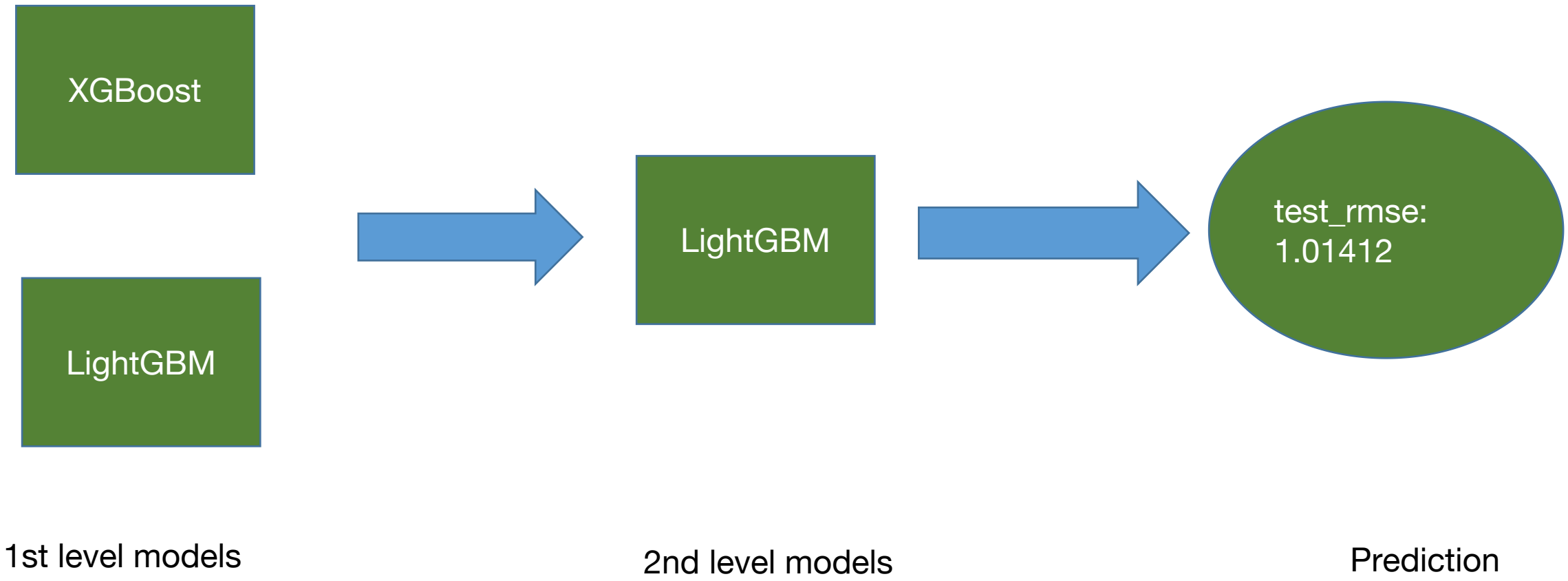
Final Values:

params['eta'] = 0.01
params['min_child_weight'] = 2
params['colsample_bytree'] = 0.8
params[‘subsample'] = 0.8
params['max_depth'] = 5
params[‘gamma'] = 0
params['lambda'] = 0.3
params['alpha'] = 0.6

Best iteration:[850]

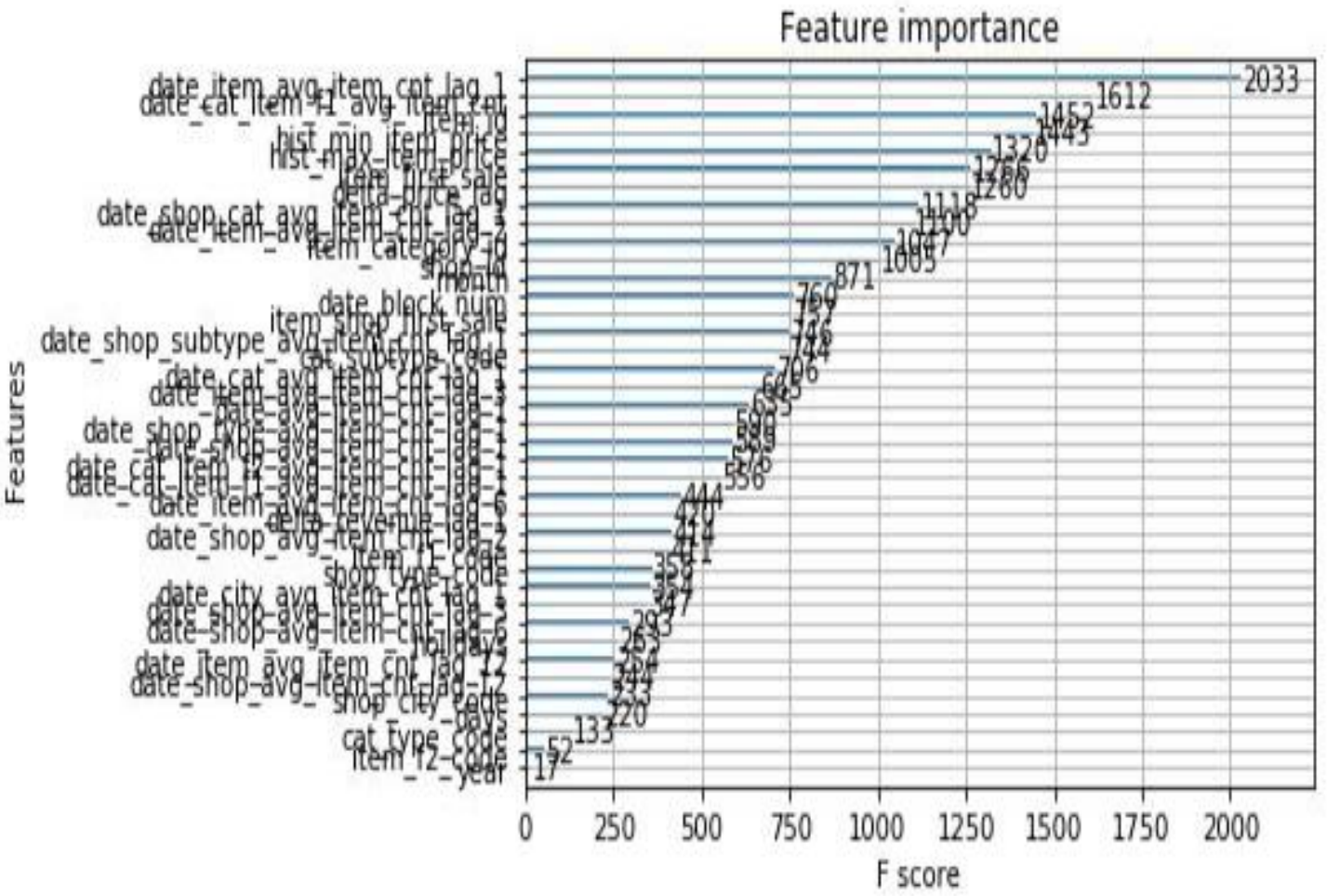
test_rmse:0.90912

Model Ensembling



04 RESULTS

RESULTS



test_rmse:0.90912

date_item_avg_item_cnt_lag_1
date_cat_item_f1_avg_item_cnt_lag_1
item_id
hist_min_item_price
hist_max_item_price
item_first_sale
delta_price_lag

05 CONCLUSION

THANKS FOR WATCHING