Predict future sales

Pengcheng Jiang

JiLin University

2021-04-19

Overview

Problem Definition

Predict future sales

- given: a challenging time-series dataset consisting of daily sales data, kindly provided by one of the largest Russian software firms - 1C Company.
- target: predict total sales for every product and store in the next month
- evaluation: Submissions are evaluated by root mean squared error (RMSE)

Data Cleaning

Date

- item_categories.csv:item_category_name item_category_id
- items.csv:item_id item_category_id
- sales_train.csv:date date_block_num shop_id item_id item_price item_cnt_day
- shops.csv:shop_name shop_id
- test.csv:shop_id item_id

Data Information

sales_train:

- 2935849 rows,6 columns
- 21807 items,60 shops
- data_type
  - data: object
  - date_block_num: int
  - shop_id:int
  - item_id:int
  - item_price:float
  - item_cnt_day:float

Data Information

test:

- 214200 rows,3 columns
- 5100 items,40 shops
- data_type
  - ID:int
  - shop_id:int
  - item_id:int

From here you can see a lot of stores, goods in training set are not in the test set

Missing Value and Non Value

target:Find out whether there are empty values or missing values in the data

result:

missing value:0

nan value:0

Data leakages

target:delete stores, goods in training set but not in the test set

result:sales_train

rows:1224439

items:4716

shops:42

Data duplication

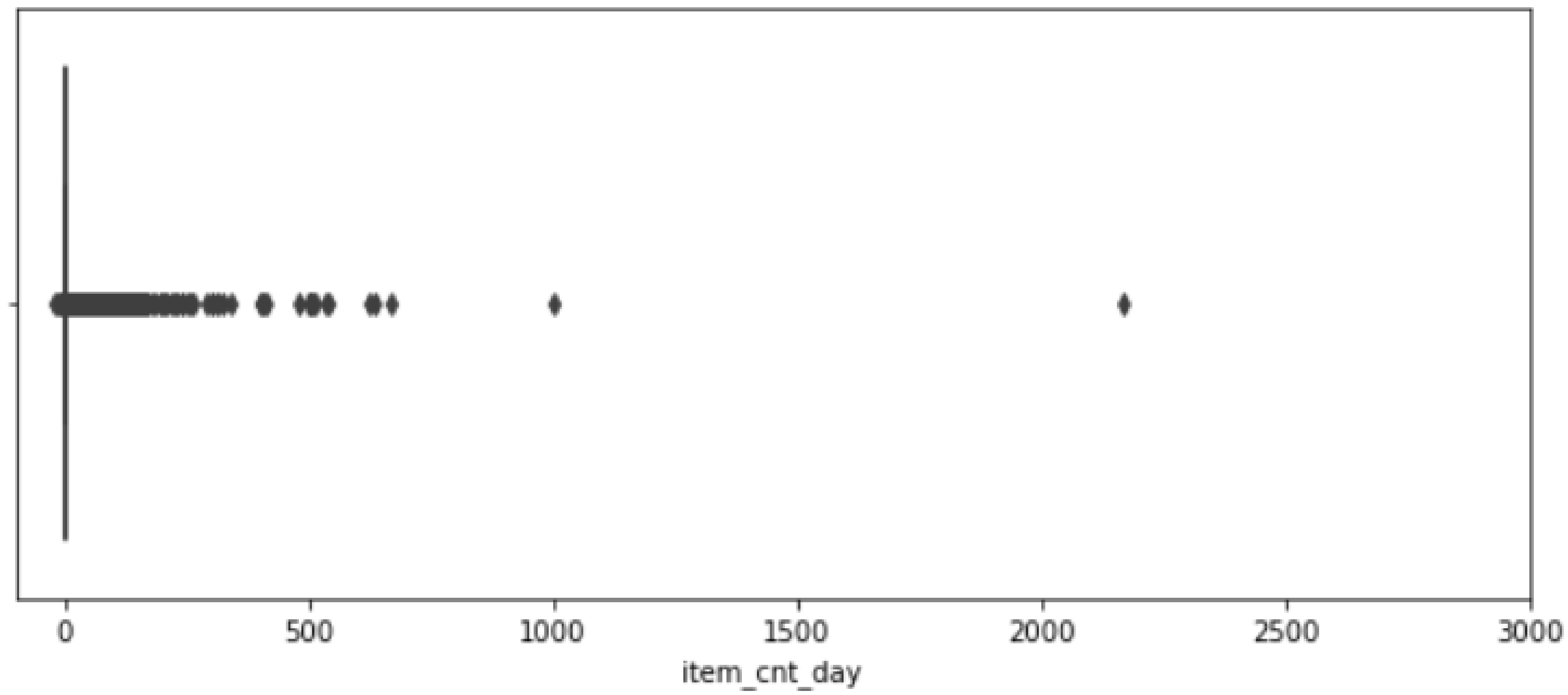
target:See if duplicate items exist in the dataset result:sales_train:0 test:0

Test set missing fill

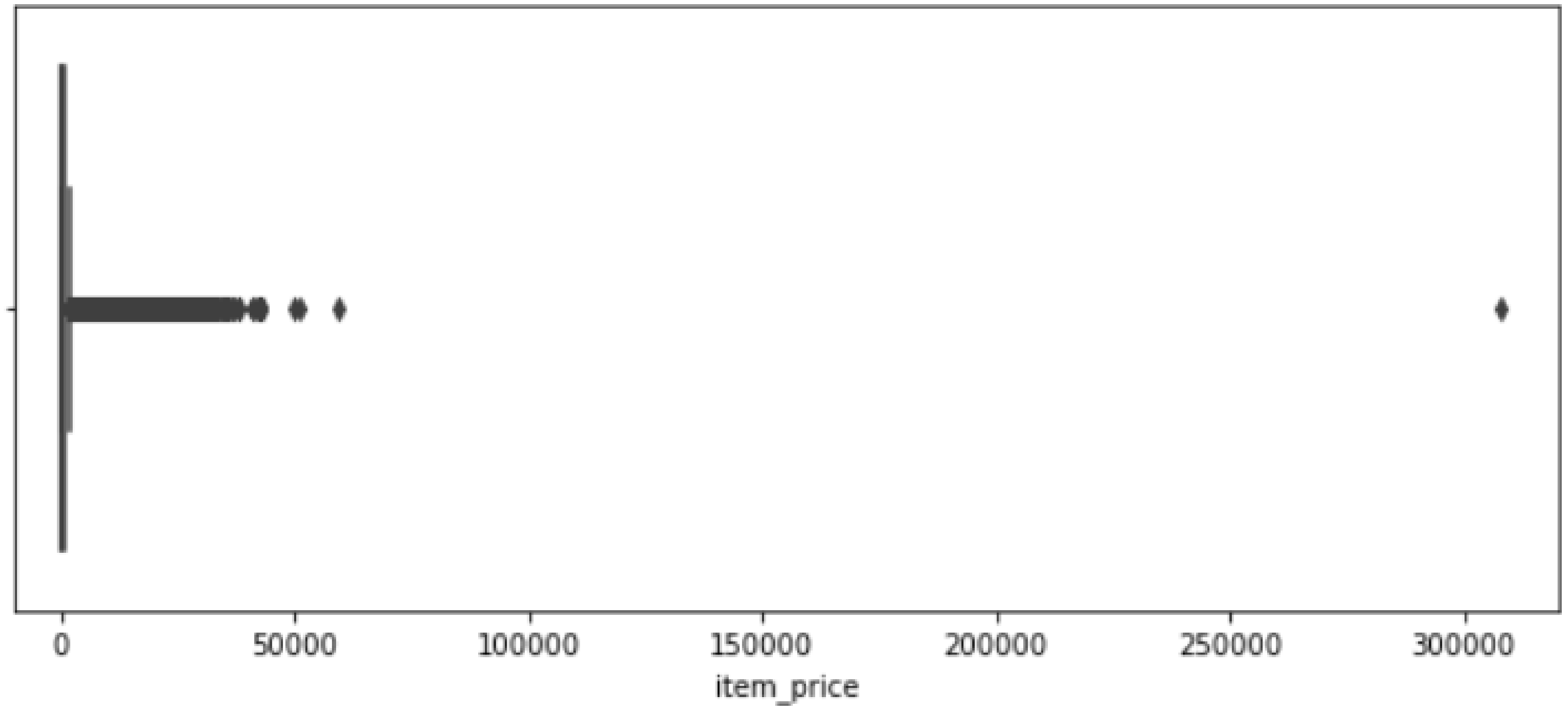
Outliers

target:Calculate the outliers of item_cnt_day and item price operation: result:

Outliers



Outliers



outdated items

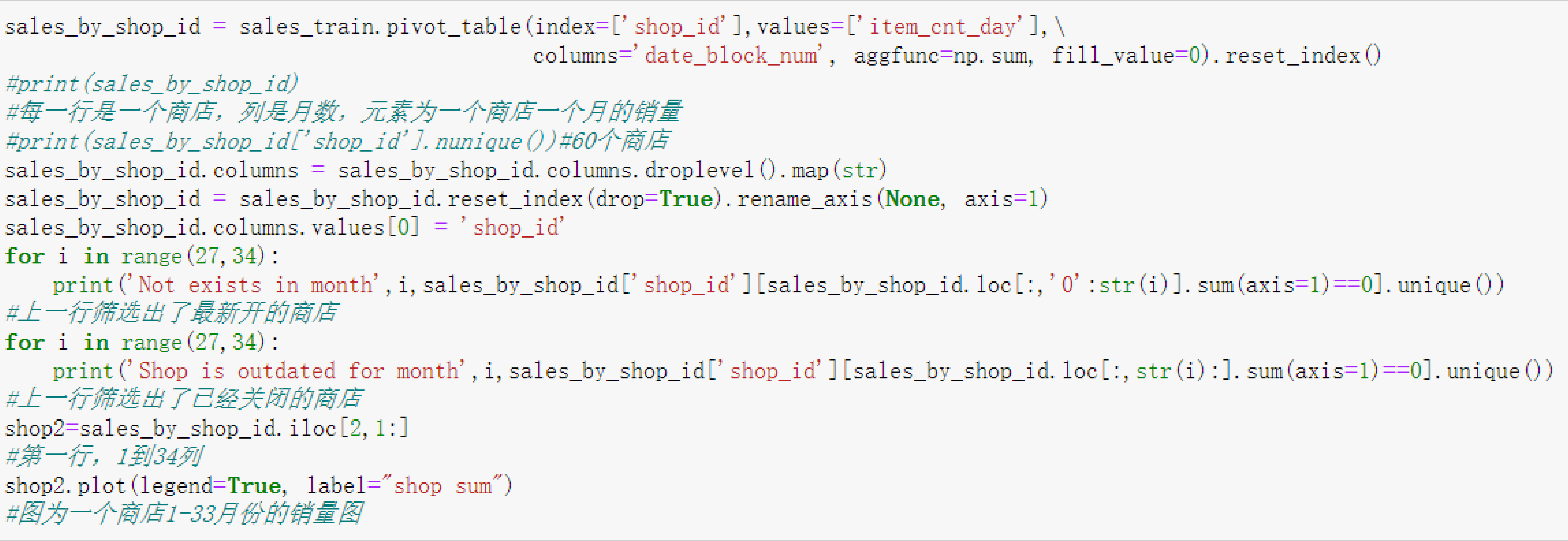
target:Analyze how many products have not been sold in the last six consecutive months. How many of these products appear in the test set. result: There are 12391 training sets, which have not been sold in the last six months. There are 164 test sets, which have not been sold in the last six months

Negative

Change item whose commodity price is negative to median operation:

Data analysis

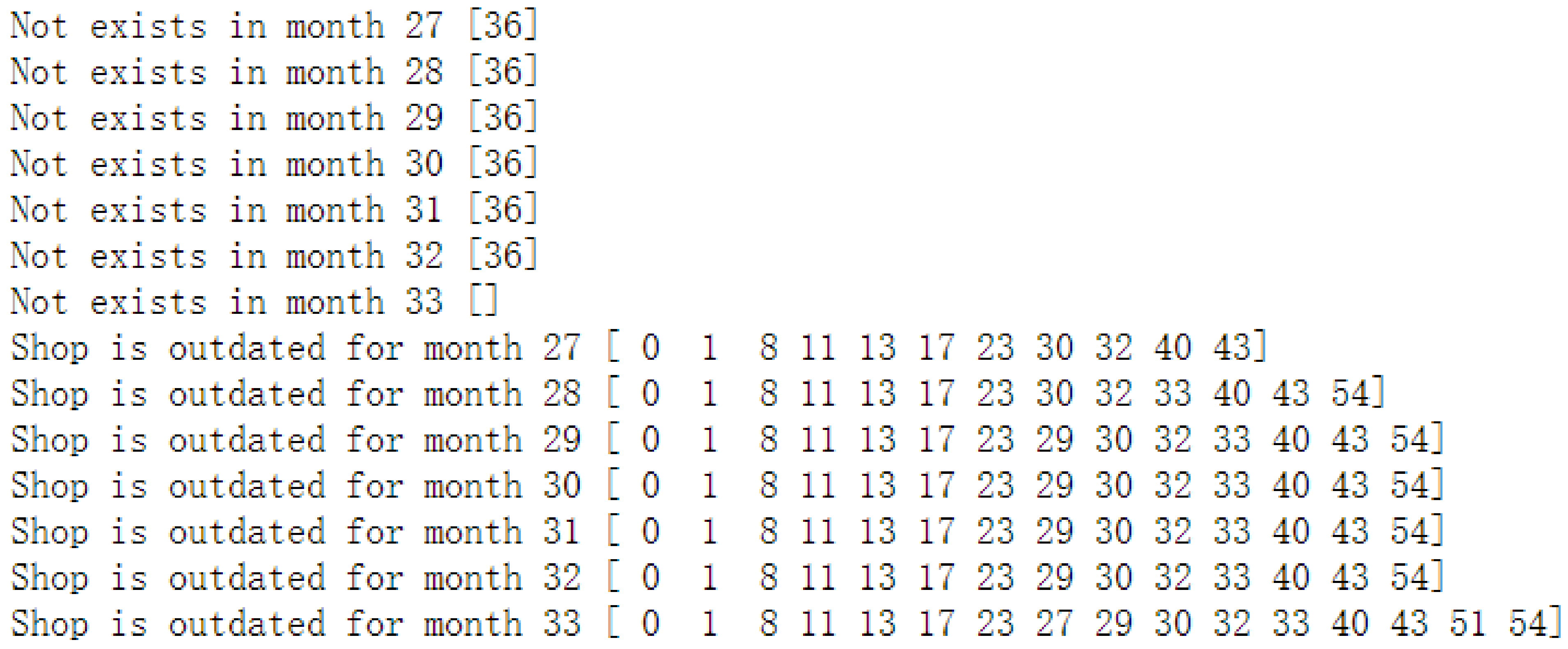
Shop sales



```
sales_by_shop_id = sales_train.pivot_table(index=['shop_id'],values=['item_cnt_day'],\
                                           columns='date_block_num', aggfunc=np.sum, fill_value=0).reset_index()
#print(sales_by_shop_id)
#每一行是一个商店，列是月数，元素为一个商店一个月的销量
#print(sales_by_shop_id['shop_id'].nunique())#60个商店
sales_by_shop_id.columns = sales_by_shop_id.columns.droplevel().map(str)
sales_by_shop_id = sales_by_shop_id.reset_index(drop=True).rename_axis(None, axis=1)
sales_by_shop_id.columns.values[0] = 'shop_id'
for i in range(27,34):
    print('Not exists in month',i,sales_by_shop_id['shop_id'][sales_by_shop_id.loc[:,'0':str(i)].sum(axis=1)==0].unique())
#上一行筛选出了最新开的商店
for i in range(27,34):
    print('Shop is outdated for month',i,sales_by_shop_id['shop_id'][sales_by_shop_id.loc[:,str(i):].sum(axis=1)==0].unique())
#上一行筛选出了已经关闭的商店
shop2=sales_by_shop_id.iloc[2,1:]
#第一行，1到34列
shop2.plot(legend=True, label="shop sum")
#图为一个商店1-33月份的销量图
```

Objective: To prepare for feature extraction

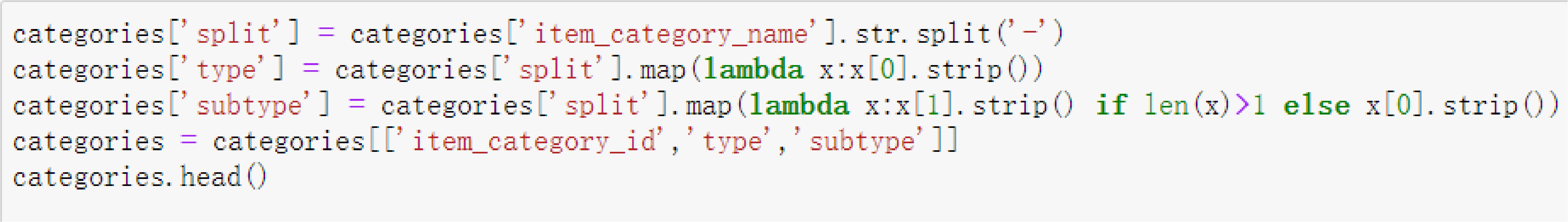
Shop sales



```
Not exists in month 27 [36]
Not exists in month 28 [36]
Not exists in month 29 [36]
Not exists in month 30 [36]
Not exists in month 31 [36]
Not exists in month 32 [36]
Not exists in month 33 []
Shop is outdated for month 27 [ 0  1  8 11 13 17 23 30 32 40 43]
Shop is outdated for month 28 [ 0  1  8 11 13 17 23 30 32 33 40 43 54]
Shop is outdated for month 29 [ 0  1  8 11 13 17 23 29 30 32 33 40 43 54]
Shop is outdated for month 30 [ 0  1  8 11 13 17 23 29 30 32 33 40 43 54]
Shop is outdated for month 31 [ 0  1  8 11 13 17 23 29 30 32 33 40 43 54]
Shop is outdated for month 32 [ 0  1  8 11 13 17 23 29 30 32 33 40 43 54]
Shop is outdated for month 33 [ 0  1  8 11 13 17 23 27 29 30 32 33 40 43 51 54]
```

Item Information

The categories of items are: large categories, small categories, we separate them, and code them separately to facilitate subsequent feature extraction

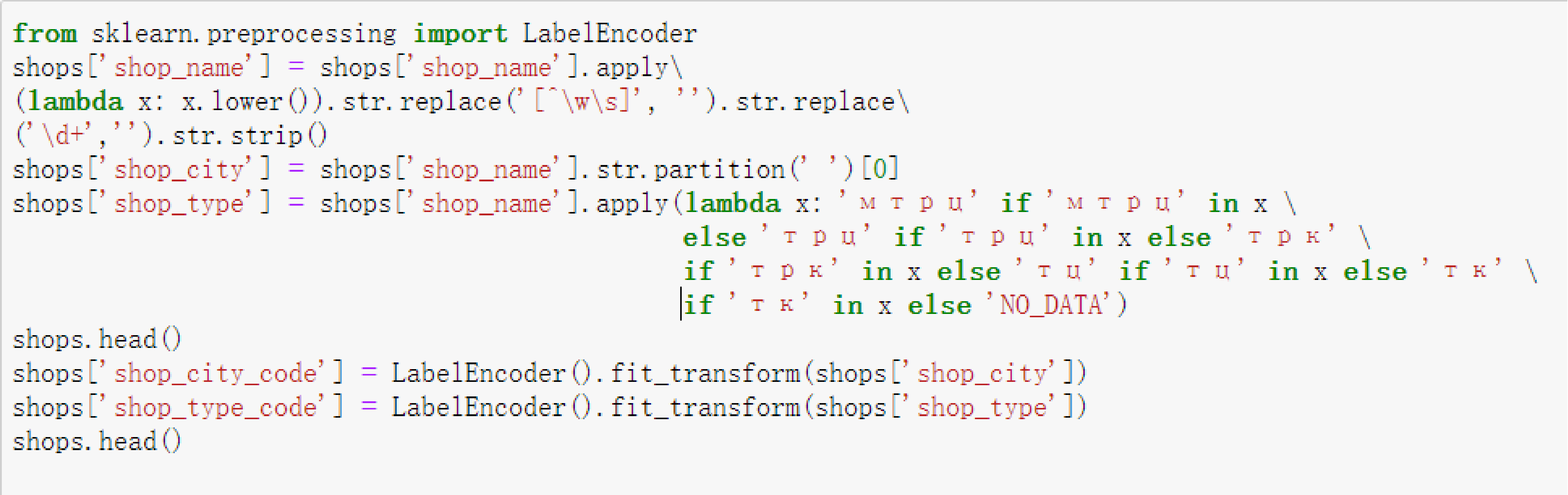


```
categories['split'] = categories['item_category_name'].str.split('-')
categories['type'] = categories['split'].map(lambda x:x[0].strip())
categories['subtype'] = categories['split'].map(lambda x:x[1].strip() if len(x)>1 else x[0].strip())
categories = categories[['item_category_id','type','subtype']]
categories.head()
```

Shop Information

Shop information includes: the city where the store is located, the type of store, which we separate and encode separately for subsequent feature extraction

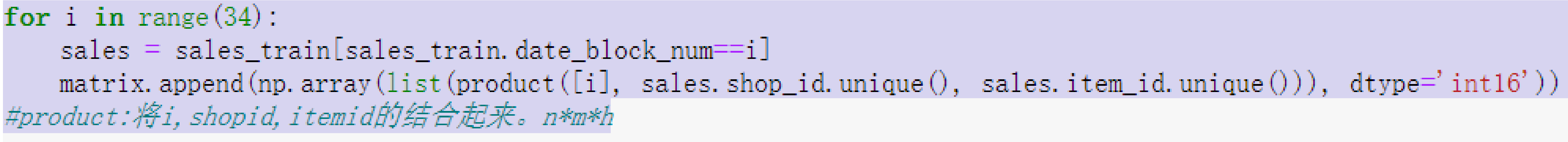
Shop Information



```
from sklearn.preprocessing import LabelEncoder
shops['shop_name'] = shops['shop_name'].apply\
(lambda x: x.lower()).str.replace('[^\w\s]', '').str.replace\
('\d+','').str.strip()
shops['shop_city'] = shops['shop_name'].str.partition(' ')[0]
shops['shop_type'] = shops['shop_name'].apply(lambda x: 'мтрц' if 'мтрц' in x \
                                              else 'трц' if 'трц' in x else 'трк' \
                                              if 'трк' in x else 'тц' if 'тц' in x else 'тк' \
                                              if 'тк' in x else 'NO_DATA')
shops.head()
shops['shop_city_code'] = LabelEncoder().fit_transform(shops['shop_city'])
shops['shop_type_code'] = LabelEncoder().fit_transform(shops['shop_type'])
shops.head()
```

Items Information

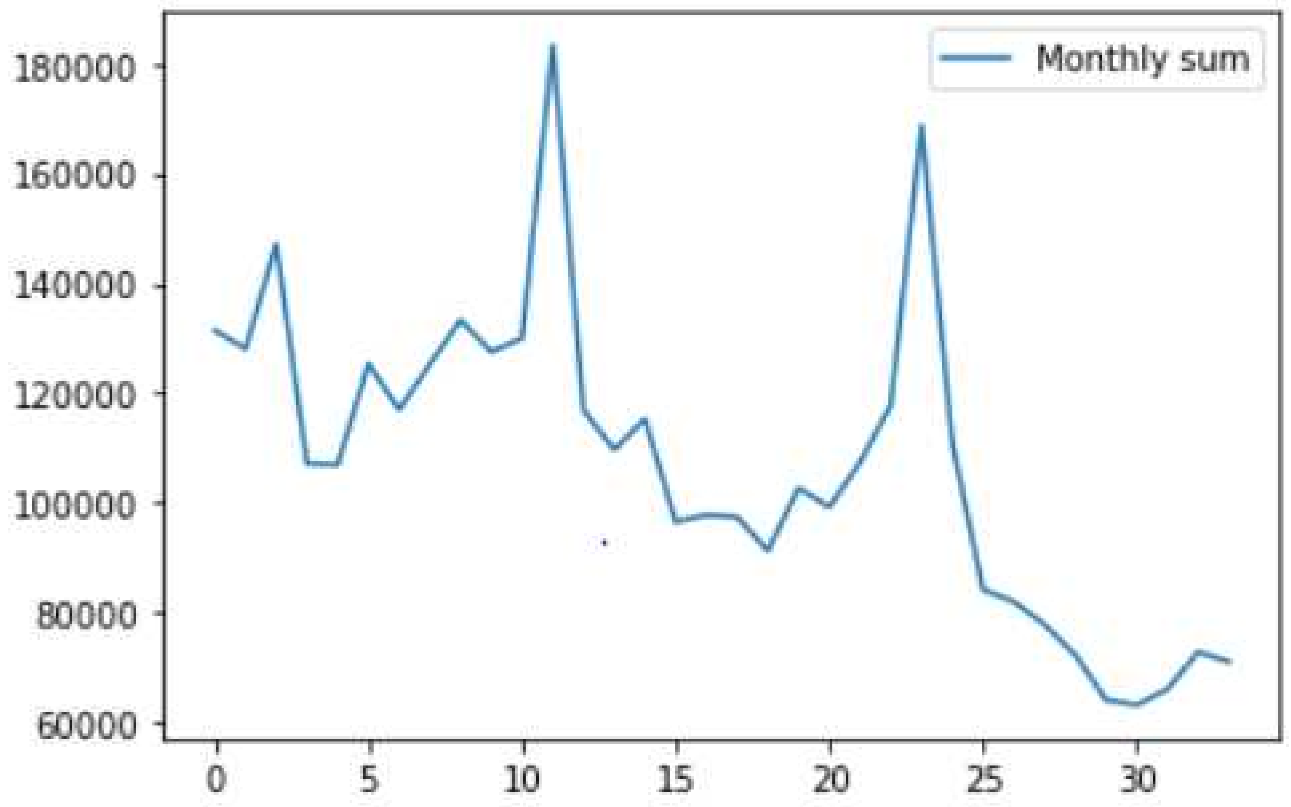
The training set contains only the items that the store actually sold that month, for items not sold during the month, you should add them and set them to 0



```
for i in range(34):
    sales = sales_train[sales_train.date_block_num==i]
    matrix.append(np.array(list(product([i], sales.shop_id.unique(), sales.item_id.unique())), dtype=’int16’))
#product:将i,shopid,itemid的结合起来。n*m*h
```

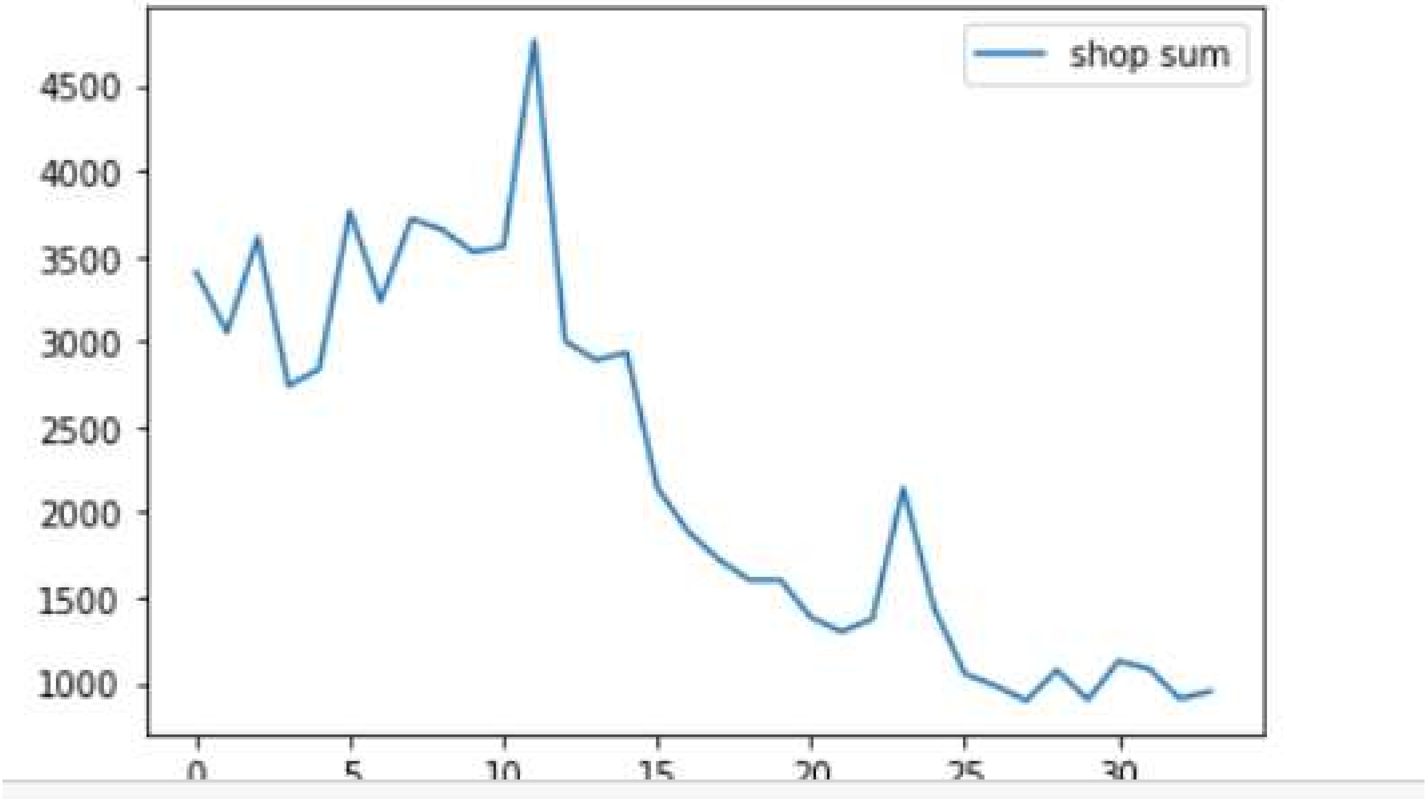
Cartesian product

Monthly total sales



Sales per store

It is known that the city to which the store belongs and the type of store affect sales



Model

Model selection

- GBDT
- Xgboost
- lightgbm
- neural network

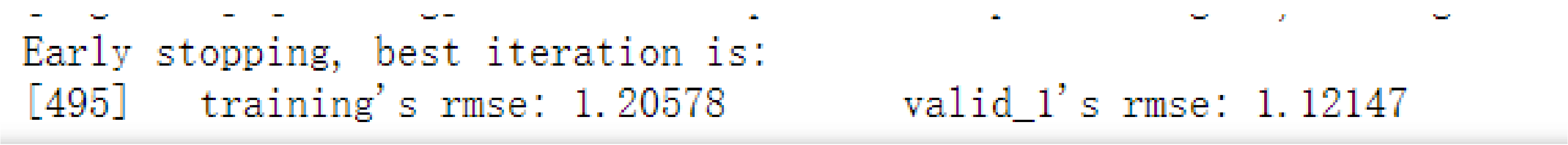
Method One

Method:The sales of the 34th month are regarded as the sales of the 35th month operation:Count the sales volume of each item in each store in the 33rd month and merge it with test Result:RMSE=1.16777

Method One

features:shop_id,item_id,item_cnt_month

Method:lightgbm



```
Early stopping, best iteration is:
[495]    training's rmse: 1.20578        valid_1's rmse: 1.12147
```

attention:After some data preprocessing

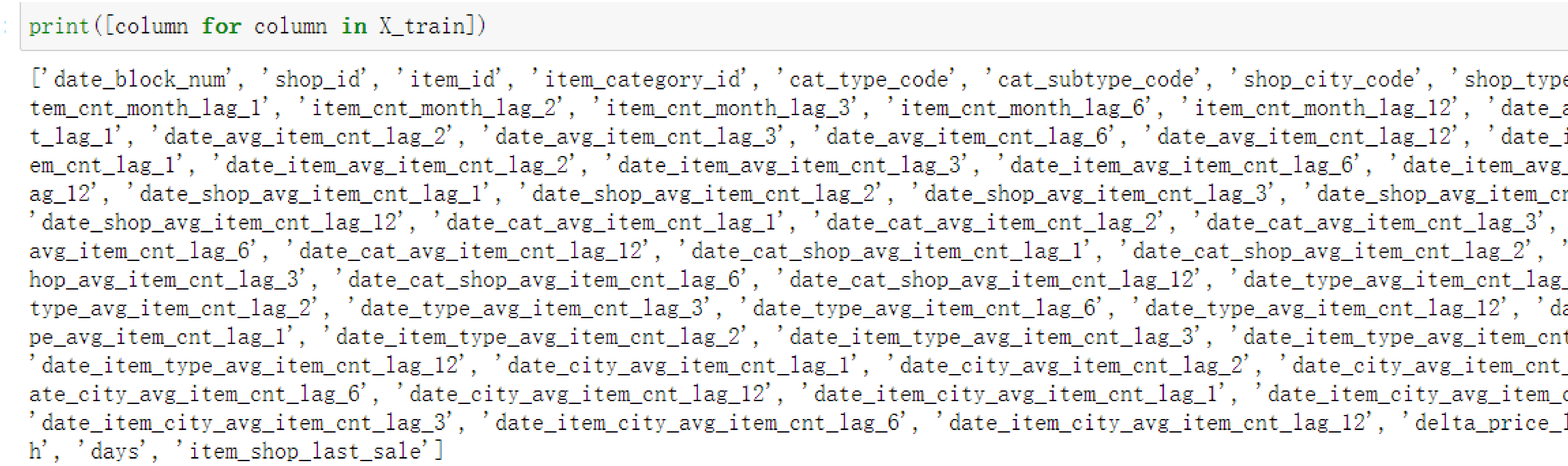
Method Two

Some historical information needs to be generated by delayed operations. For example, you can use the 0-33 month sales as a historical feature of the 1-34 month (one month delay).

Method Two

- Historical information on monthly sales (per item-store).
- Historical information on the average monthly sales (all merchandise-store) value
- Average monthly sales (per item) and historical characteristics
- Average monthly sales (per store) and historical characteristics
- Average monthly sales (per commodity category) and historical characteristics
- Average monthly sales (commodity category-store) and historical characteristics
- Average and historical characteristics of monthly sales volume (commodity category _ class)
- Average and historical characteristics of monthly sales (commodity-commodity category _ class)
- Average monthly sales (store _ city) and historical characteristics
- Average monthly sales (merchandise-store-city) and historical characteristics
- Trends, price changes over the past six months
- Number of days per month
- Sales beginning and ending

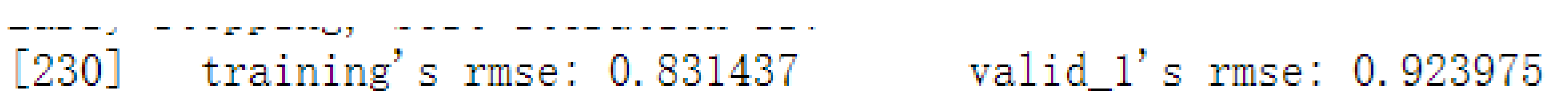
Method Two



```
print([column for column in X_train])
```

```
['date_block_num', 'shop_id', 'item_id', 'item_category_id', 'cat_type_code', 'cat_subtype_code', 'shop_city_code', 'shop_type
tem_cnt_month_lag_1', 'item_cnt_month_lag_2', 'item_cnt_month_lag_3', 'item_cnt_month_lag_6', 'item_cnt_month_lag_12', 'date_a
t_lag_1', 'date_avg_item_cnt_lag_2', 'date_avg_item_cnt_lag_3', 'date_avg_item_cnt_lag_6', 'date_avg_item_cnt_lag_12', 'date_i
em_cnt_lag_1', 'date_item_avg_item_cnt_lag_2', 'date_item_avg_item_cnt_lag_3', 'date_item_avg_item_cnt_lag_6', 'date_item_avg_
ag_12', 'date_shop_avg_item_cnt_lag_1', 'date_shop_avg_item_cnt_lag_2', 'date_shop_avg_item_cnt_lag_3', 'date_shop_avg_item_cn
'date_shop_avg_item_cnt_lag_12', 'date_cat_avg_item_cnt_lag_1', 'date_cat_avg_item_cnt_lag_2', 'date_cat_avg_item_cnt_lag_3',
avg_item_cnt_lag_6', 'date_cat_avg_item_cnt_lag_12', 'date_cat_shop_avg_item_cnt_lag_1', 'date_cat_shop_avg_item_cnt_lag_2', '
hop_avg_item_cnt_lag_3', 'date_cat_shop_avg_item_cnt_lag_6', 'date_cat_shop_avg_item_cnt_lag_12', 'date_type_avg_item_cnt_lag
type_avg_item_cnt_lag_2', 'date_type_avg_item_cnt_lag_3', 'date_type_avg_item_cnt_lag_6', 'date_type_avg_item_cnt_lag_12', 'da
pe_avg_item_cnt_lag_1', 'date_item_type_avg_item_cnt_lag_2', 'date_item_type_avg_item_cnt_lag_3', 'date_item_type_avg_item_cnt
'date_item_type_avg_item_cnt_lag_12', 'date_city_avg_item_cnt_lag_1', 'date_city_avg_item_cnt_lag_2', 'date_city_avg_item_cnt_
ate_city_avg_item_cnt_lag_6', 'date_city_avg_item_cnt_lag_12', 'date_item_city_avg_item_cnt_lag_1', 'date_item_city_avg_item_c
'date_item_city_avg_item_cnt_lag_3', 'date_item_city_avg_item_cnt_lag_6', 'date_item_city_avg_item_cnt_lag_12', 'delta_price_l
h', 'days', 'item_shop_last_sale']
```

Method Two



```
Early stopping, best iteration is:
[230]	training's rmse: 0.831437	valid_1's rmse: 0.923975
```

Lightgbm