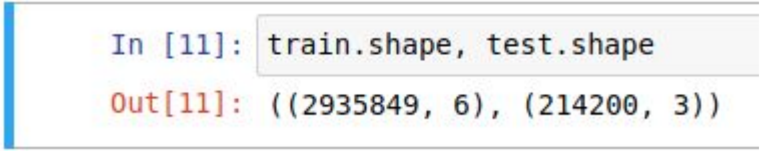
Predict future sales

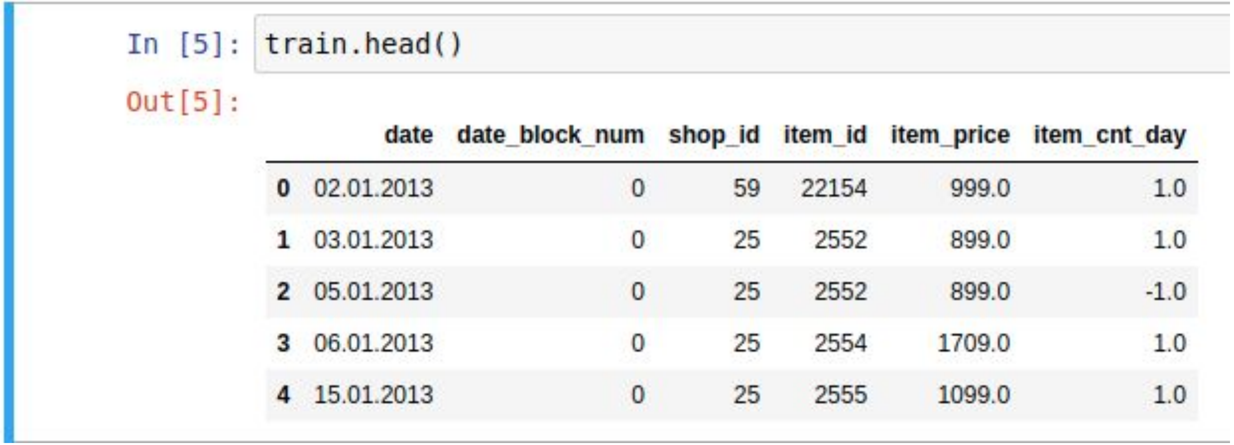
Kaggle competition
https://www.kaggle.com/c/competitive-data-science-predict-future-sales/data

Goal

- Predict total number of sales for every product and store in the next month
  - given daily sales data provided by company 1C Company
- Time series dataset, time period-33 months

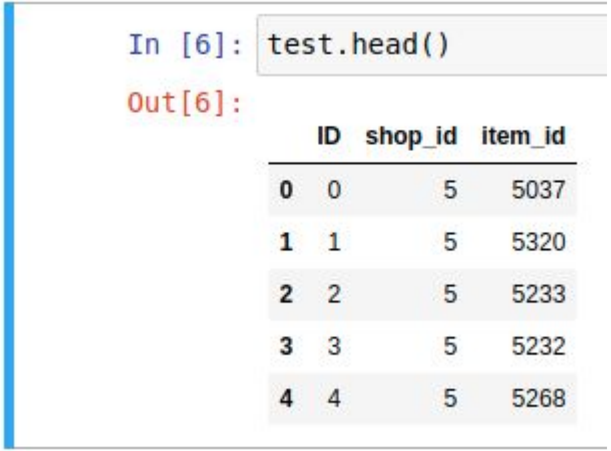


```
In [11]: train.shape, test.shape
Out[11]: ((2935849, 6), (214200, 3))
```



```
In [5]: train.head()
Out[5]:
```

	date	date_block_num	shop_id	item_id	item_price	item_cnt_day
0	02.01.2013	0	59	22154	999.0	1.0
1	03.01.2013	0	25	2552	899.0	1.0
2	05.01.2013	0	25	2552	899.0	-1.0
3	06.01.2013	0	25	2554	1709.0	1.0
4	15.01.2013	0	25	2555	1099.0	1.0

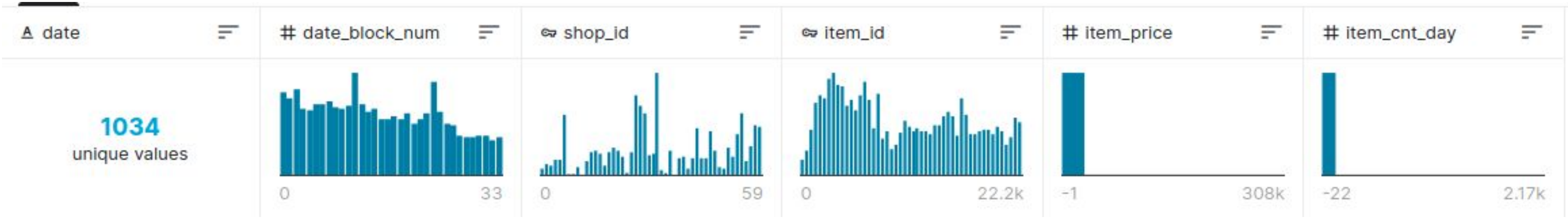


```
In [6]: test.head()
Out[6]:
```

	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

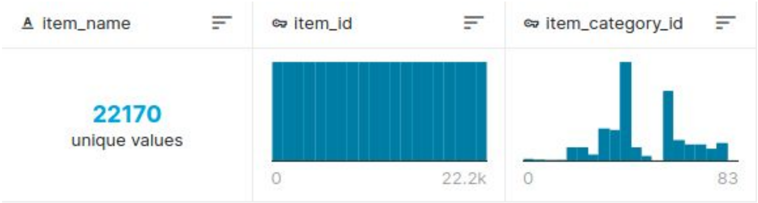
Data

Sales



date	date_block_num	shop_id	item_id	item_price	item_cnt_day
1034 unique values					

Items



item_name	item_id	item_category_id
22170 unique values		

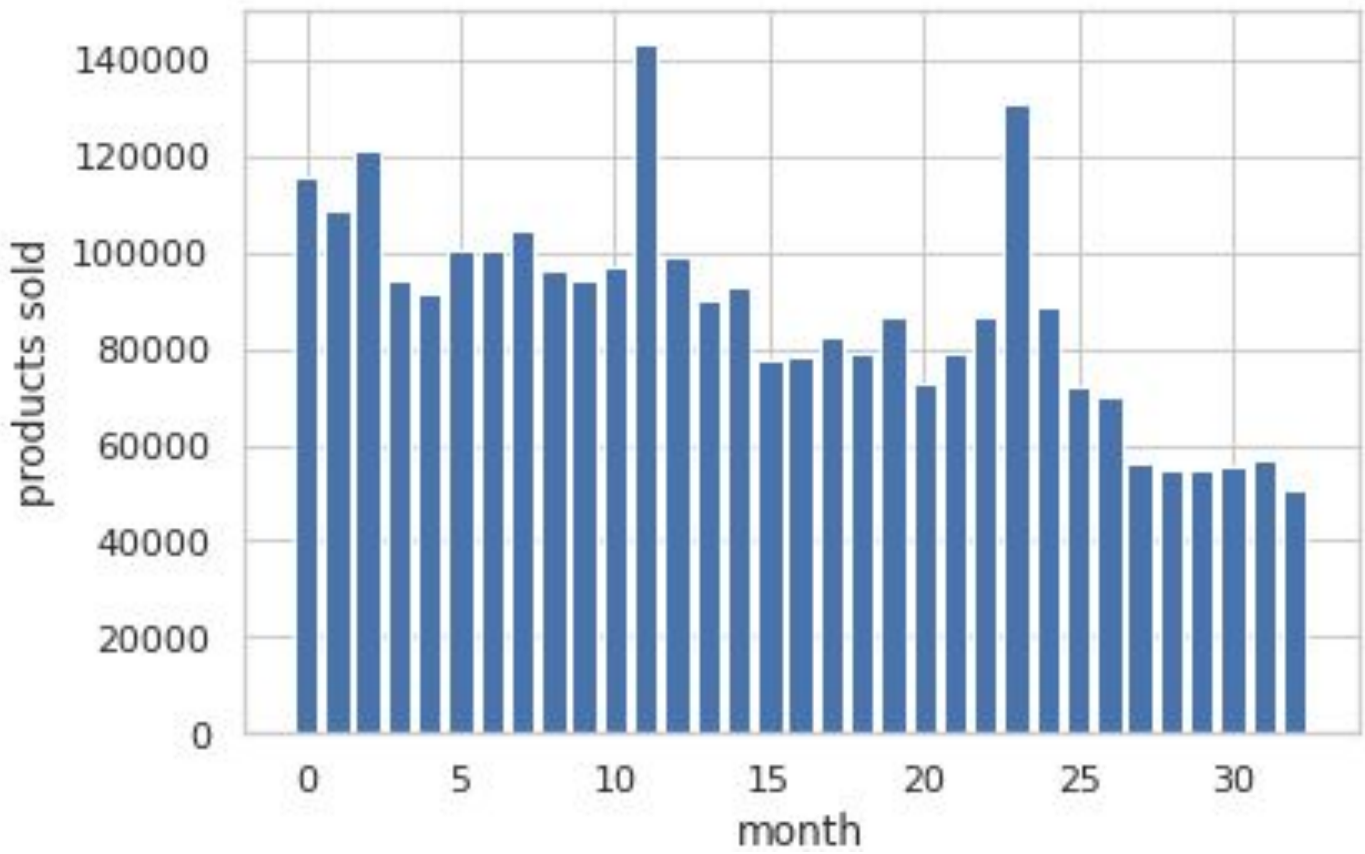
Categories

84

unique values

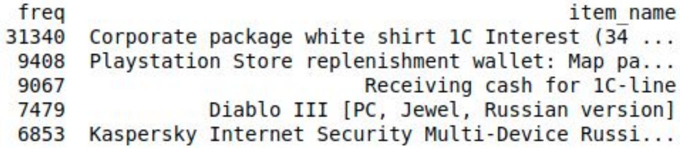
Data exploration

- Sales by month

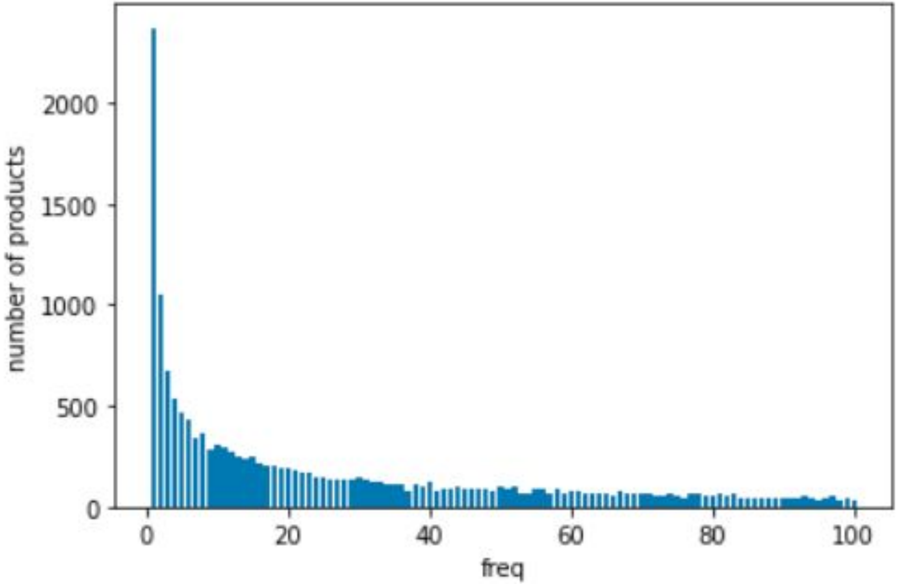


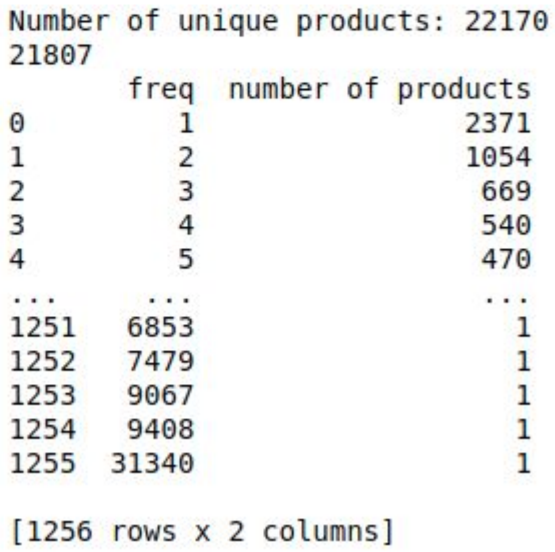
Data exploration

- Product marketability
- 2371 products that have been sold once



```
 freq                                              item_name
31340  Corporate package white shirt 1C Interest (34 ...
 9408  Playstation Store replenishment wallet: Map pa...
 9067                           Receiving cash for 1C-line
 7479             Diablo III [PC, Jewel, Russian version]
 6853  Kaspersky Internet Security Multi-Device Russi...
```



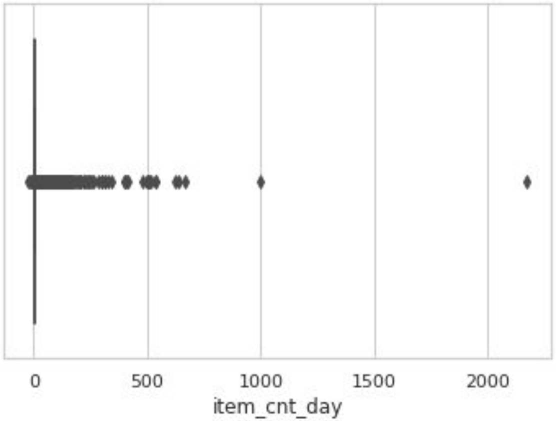


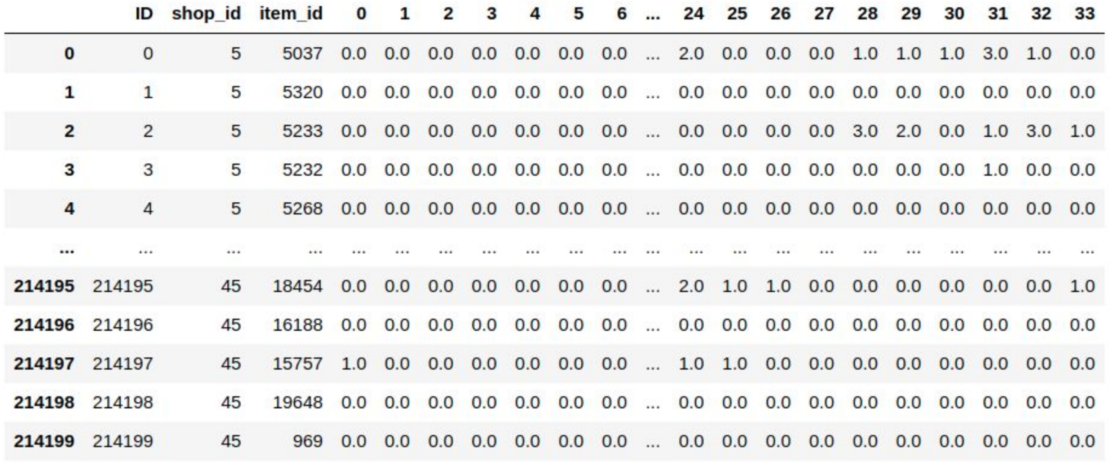
```
Number of unique products: 22170
21807
       freq  number of products
0         1                2371
1         2                1054
2         3                 669
3         4                 540
4         5                 470
...     ...                 ...
1251   6853                   1
1252   7479                   1
1253   9067                   1
1254   9408                   1
1255  31340                   1

[1256 rows x 2 columns]
```

Preprocessing

1. Daily sale records -> monthly sale records
   - ○ we are supposed to predict the total number of product sales in next month
2. Removing outliers
3. Negative sales (product returns)
4. Constructing additional features
5. Validation set

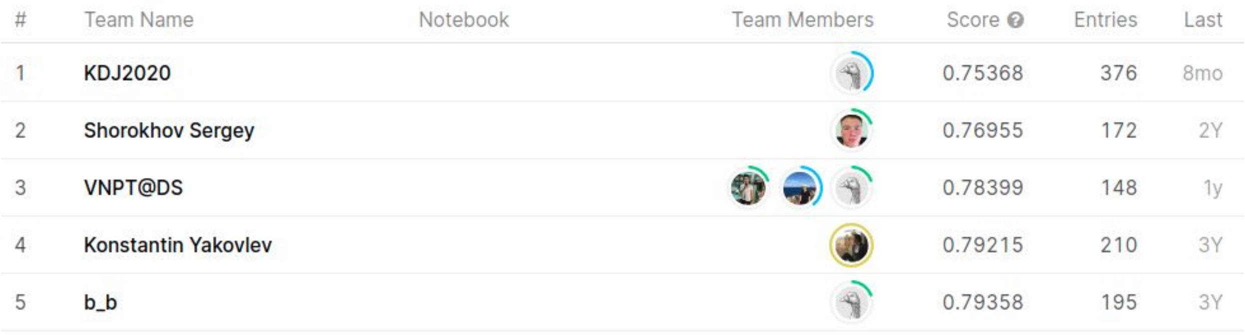




	ID	shop_id	item_id	0	1	2	3	4	5	6	...	24	25	26	27	28	29	30	31	32	33
0	0	5	5037	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	2.0	0.0	0.0	0.0	1.0	1.0	1.0	3.0	1.0	0.0
1	1	5	5320	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	2	5	5233	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	3.0	2.0	0.0	1.0	3.0	1.0
3	3	5	5232	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
4	4	5	5268	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
214195	214195	45	18454	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	2.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
214196	214196	45	16188	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
214197	214197	45	15757	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
214198	214198	45	19648	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
214199	214199	45	969	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Results

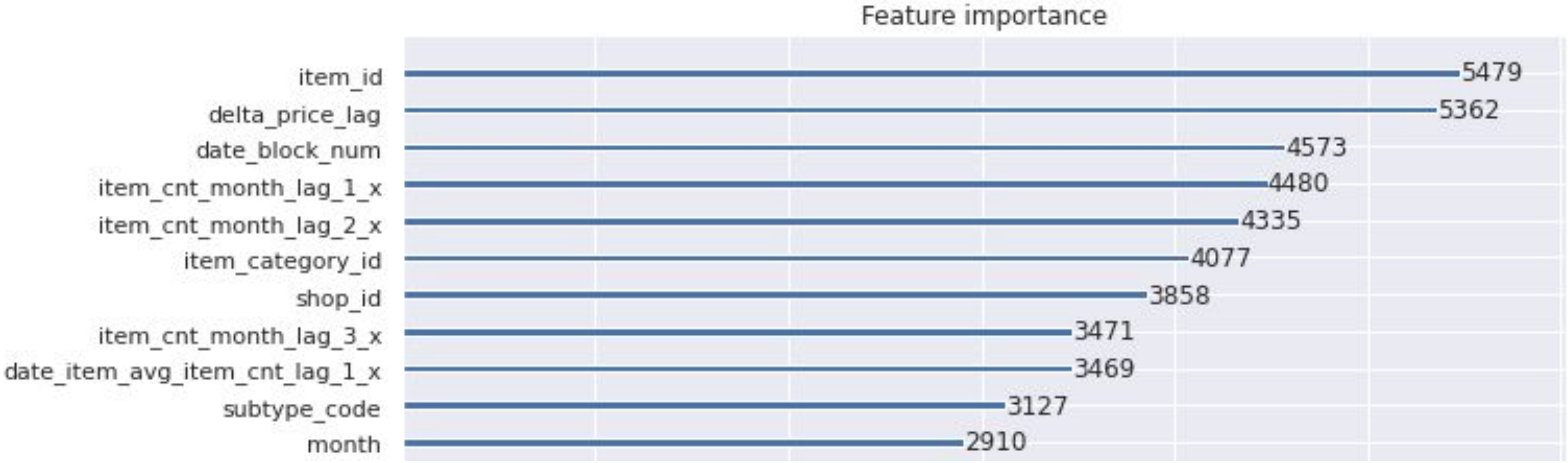
- Models
  - LGBMRegressor
    - 0.88
  - XGBRegressor
    - 1.02
  - RNN (PyTorch)
    - 0.83
- Score - Root mean squared error



#	Team Name	Notebook	Team Members	Score	Entries	Last
1	KDJ2020			0.75368	376	8mo
2	Shorokhov Sergey			0.76955	172	2Y
3	VNPT@DS			0.78399	148	1y
4	Konstantin Yakovlev			0.79215	210	3Y
5	b_b			0.79358	195	3Y

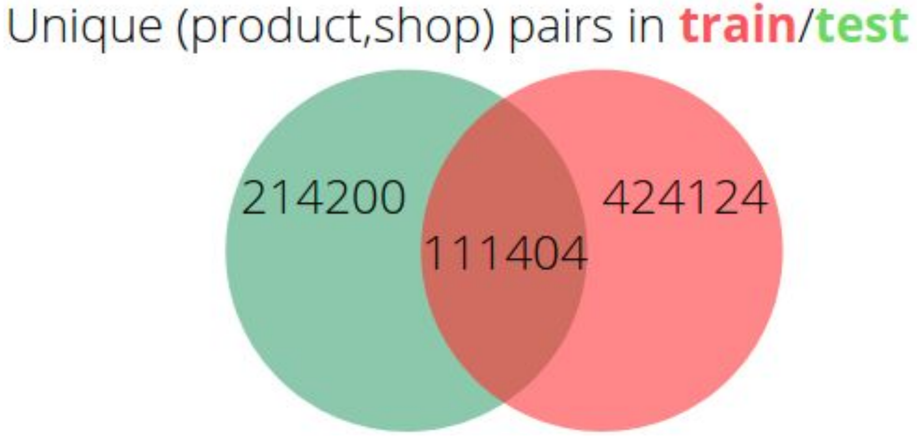
Feature engineering

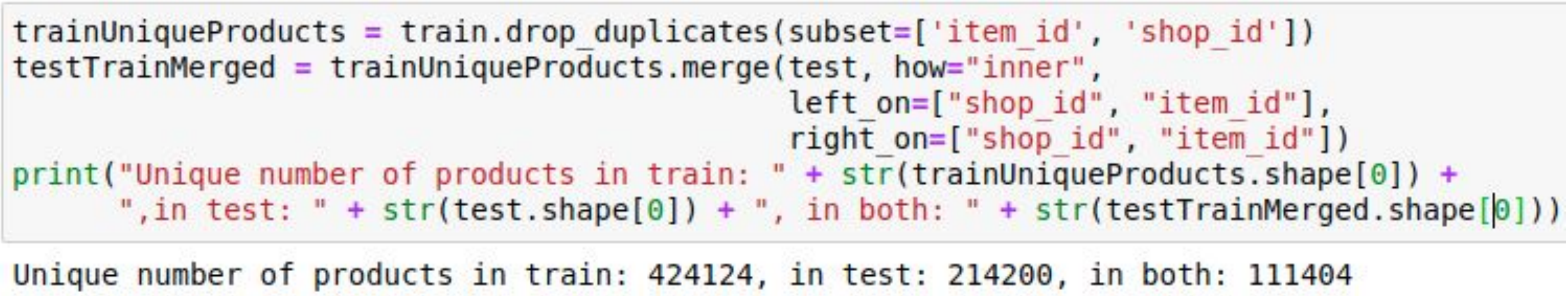
- Lag features
- Delta_price_lag



Issue - missing training samples

- We have no sale history in the train set for half of the products that we have to predict
- We can try to generate some representative values
  - Cannot validate these values


Unique (product,shop) pairs in train/test


```
trainUniqueProducts = train.drop_duplicates(subset=['item_id', 'shop_id'])
testTrainMerged = trainUniqueProducts.merge(test, how="inner",
                                            left_on=["shop_id", "item_id"],
                                            right_on=["shop_id", "item_id"])
print("Unique number of products in train: " + str(trainUniqueProducts.shape[0]) +
      ",in test: " + str(test.shape[0]) + ", in both: " + str(testTrainMerged.shape[0]))
```

```
Unique number of products in train: 424124, in test: 214200, in both: 111404
```

Solution?

- Idea
  - For every pair <item_id, shop_id> not included in train set, do:
    1. Mean of all pairs <item_id, X>
    2. If there are no pairs <item_id, X> in train set (exactly 15246 cases), do:
       - Mean of all pairs <Y, shop_id>
       - In case there are no pairs <Y, shop_id> (0 cases) in train set set the value to zero
    3. Use this value as a prediction

- Did it work? - Not really
- Worse results

Things worth trying

- CatBoost
- AutoML
- Additional feature extraction
- Still plenty of time to submit: **Competition ends in Jan 1. 2023**

Thanks for your attention