January 6, 2019
Krzysztof Joachimiak
joachimiak.krzysztof@gmail.com
github.com/krzjoa/kaggle-sales

Pearson

Predict Future Sales - Kaggle competition

Recruitment task for Research Engineer position

Contents

1 Task

The goal of this task is to predict future sales value. This task is a Kaggle competition.

2 Data Analysis

2.1 General information

There are **22170** divided into **84 categories**. In the dataset, there occur **60 shops**. We can find **2'935'849 records** in the training dataset, and **214'200** in the testing one.

Insigths:

- There occur **negative coun values**. As many guys in the competition-related discussion say, it probably expresses the number of returned and refunded items

2.2 Trends in Time Series

At the very beginning, let's check, how many recordings per each month in the measured period we have. As we can see in the figure 2.1, the number of sale records depends on time and we are not sure if it's just a **lack of data** or it really shows us some **meaningful temporal relation**.

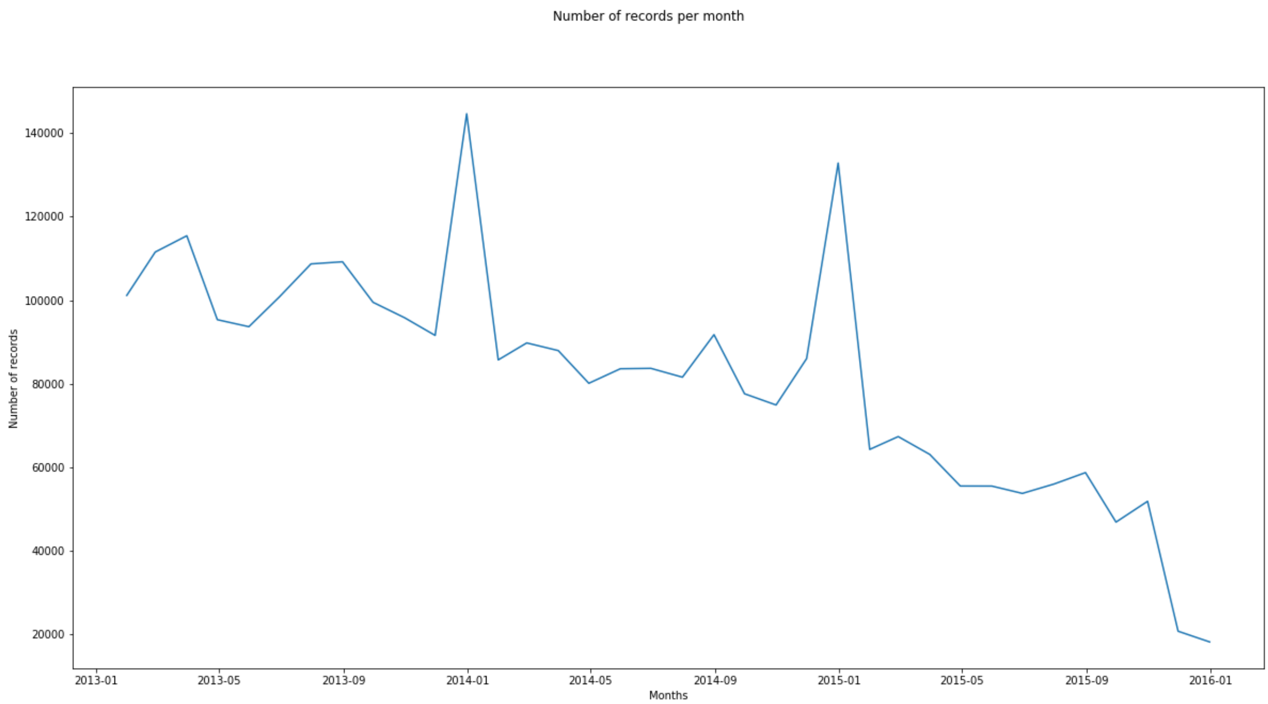

Figure 2.1: Number of records per month in the training dataset

If we sum values of sales in the each month, we can get following plot (figure 2.2).

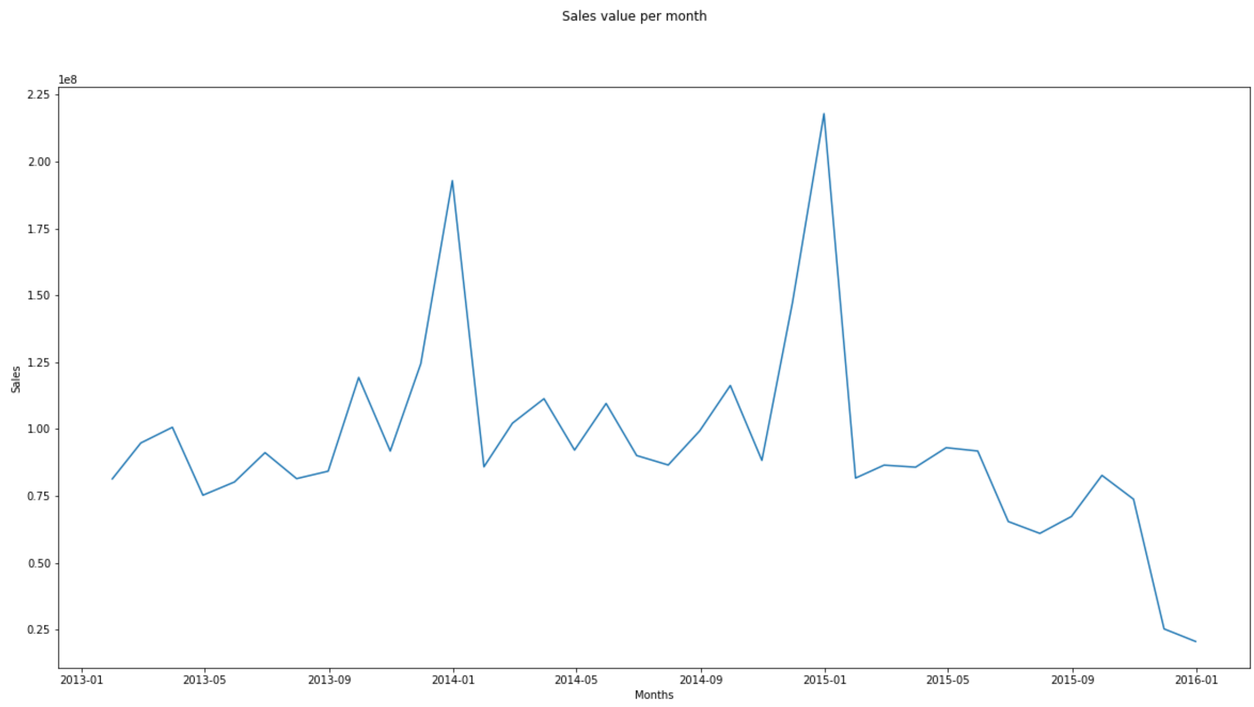

Figure 2.2: Sales values per month in the training dataset

Seeing only this picture, we cannot state if there exists a clear link between time and summed sales values in each month.

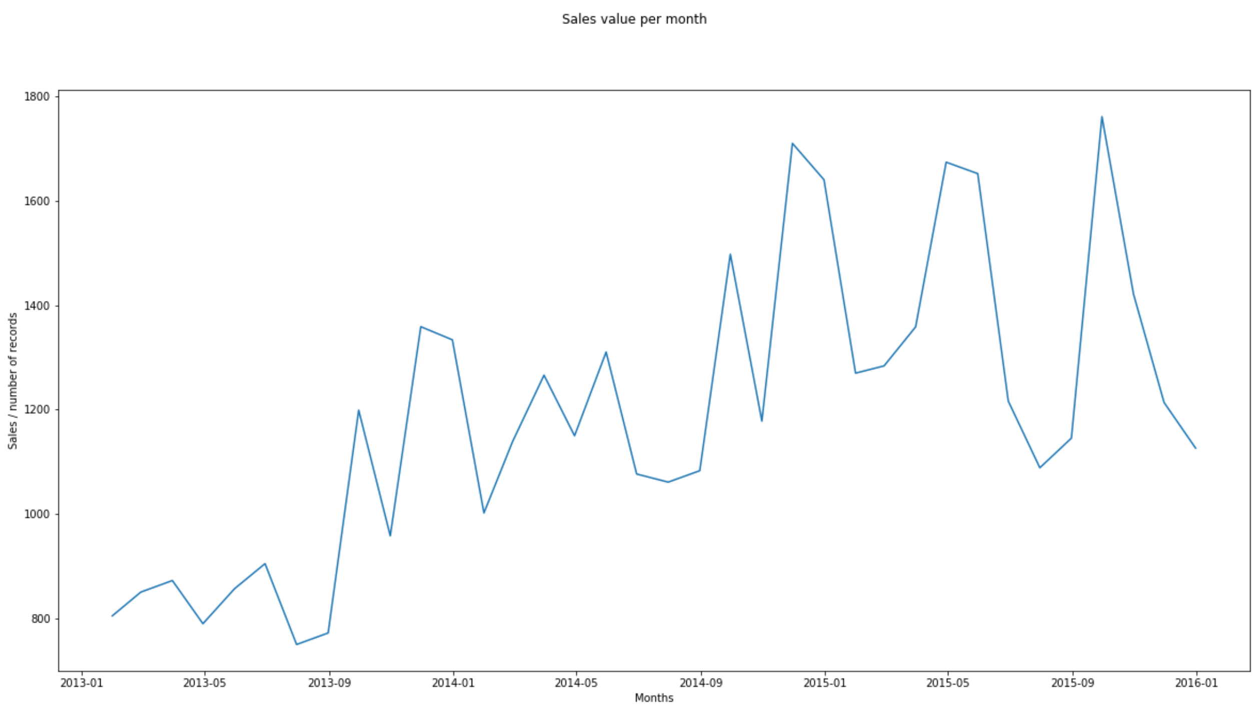

Figure 2.3: Sales values per month in the training dataset

Bearing in mind the previous plot (fig. 2.1), we can apply some kind of normalization and check, how does the temporal relation between sales and number of records looks like. It is presented in the figure 2.3. It suggest, that some increasing trend may exist.

Analyzing sales changes in some specific seasons is useless because we are not given such information in the our testing dataset.

2.3 Shops

2.4 Categories

3 Features

3.1 features_v1

This is a simpliefied feature subset, constrained only to plain use of *shop_id* and *item_id* as categorical features.

3.2 features_v2

Using code in *Feature engineering - features_v2* notebook, new features were added to each record to both *sales_train_v2.csv* and *test.csv* files. Full set of features is as follows:

- shop_id
- item_id
- total_cat_cnt - total number of items sold in the category
- min_cat_cnt
- max_cat_cnt
- mean_cat_cnt
- std_cat_price
- min_cat_price
- max_cat_price
- mean_cat_price
- std_cat_price
- total_shop_cnt
- min_shop_cnt
- max_shop_cnt
- mean_shop_cnt
- std_shop_price
- min_shop_price
- max_shop_price
- mean_shop_price
- std_shop_price

4 Score

In order to measure efficiency of the given algorithm, I used the same measure, which is used in the Kaggle leaderboard, i.e. **root mean square error**. This metric is commonly used in regression tasks.

5 Experiments

5.1 Baseline

Here, I will present a simple baseline solution, which will be used as a reference for the further trials. Because of the presence of catergorical features, I used one of Gradient Boosting implementation, i.e. CatBoost by Yandex. I left all hyperparameters set by default. Baselin model code can be found in the **CatBoost - baseline.ipynb** file.

Dataset This a very simplified experiment, so here I drop temporal order of the records and divide trainig dataset into two subsets:

- training - 80% of records
- testing - 20% of records

  Feature vectors contains only a simple information about shop id and item id. It potentialy may cause problems because of change

Results Mean root mean square error after 25 trials is about 2.54.

Kaggle score Afer training on whole dataset, baseline solution gained **1.43427** on Kaggle leaderboard (place: 1692/2055).

5.2 Nonlinear regression on feature_v2

Next step is a simple neural network, which facilitates us to train a nonlinear regression model.

Dataset In this case, model efficiency was checked using well-known **cross-validation procedure**. A **five-fold split** was generated and saved as a json file. Sample indices in the dataset were randomly shuffled, so we still don't make any use of temperal relation between records. Feature set - see section 3.2

Model

Results