PREDICT FUTURE SALES

Kaggle Competition

Hubert Joskowiak

Poznan, 2018

Contents

1. Project description

- Input data: daily historical data from January 2013 to October 2015.
- Goal: forecast the total amount of products sold in every shop for the test set.
- File descriptions:
  - *sales_train.csv* - the training set. Daily historical data from January 2013 to October 2015,
  - *test.csv* - the test set,
  - *items.csv* - supplemental information about the items/products,
  - *item_categories.csv* - supplemental information about the items' categories,
  - *shops.csv* - supplemental information about the shops.

2. Proposed solution

- Use of all shops/items combination from test set as input data:
  - Features: item, shop, items' category and sales amount from previous months.
  - Labels: number of items sold every month in a certain shop.
- Use of machine learning algorithms:
  - XGBoost
  - Multi-layers neural networks
- Baseline: sales infomation from October 2015. Values were clipped to the range [0; 20]. RMSE value for test set is equal to **1,16811**
- Project goal: get RMSE value about 1,00

2.1. Kernels used

- Initial data wrangling (features extraction), baseline
  www.kaggle.com/szhou42/predict-future-sales-top-11-solution
- Correct data mistakes
  www.kaggle.com/dlarionov/feature-engineering-xgboost
- XGBoost model
  www.kaggle.com/jamesguo89/sklearn

2.2. Libraries used

- numpy
- pandas
- matplotlib.pyplot
- sklearn
- itertools
- tensorflow
- xgboost

3. Initial data wrangling

3.1. Data pre-processing

1. Loading data and building a grid with all test set item/shops list from every month
2. Deleting duplicates
3. Anomaly detection (fig 3.1)
4. Correcting anomalies:
   - item price equals to 0 - use median of prices of the same item ($\mathit{item\_id} = 2973$) and the same shop ($\mathit{shop\_id} = 32$) get from 4th month ($\mathit{date\_block\_num} = 4$)
   - item price clipped to range [0; 300000] - get rid of negative values
   - item sales clipped to range [0; 1000] - get rid of negative values and values that are very different from the average

- changing shop names and shop IDs because there are several typos; for example shops with ID numbers 10 and 11 are the same one; after the change there are 57 shops.

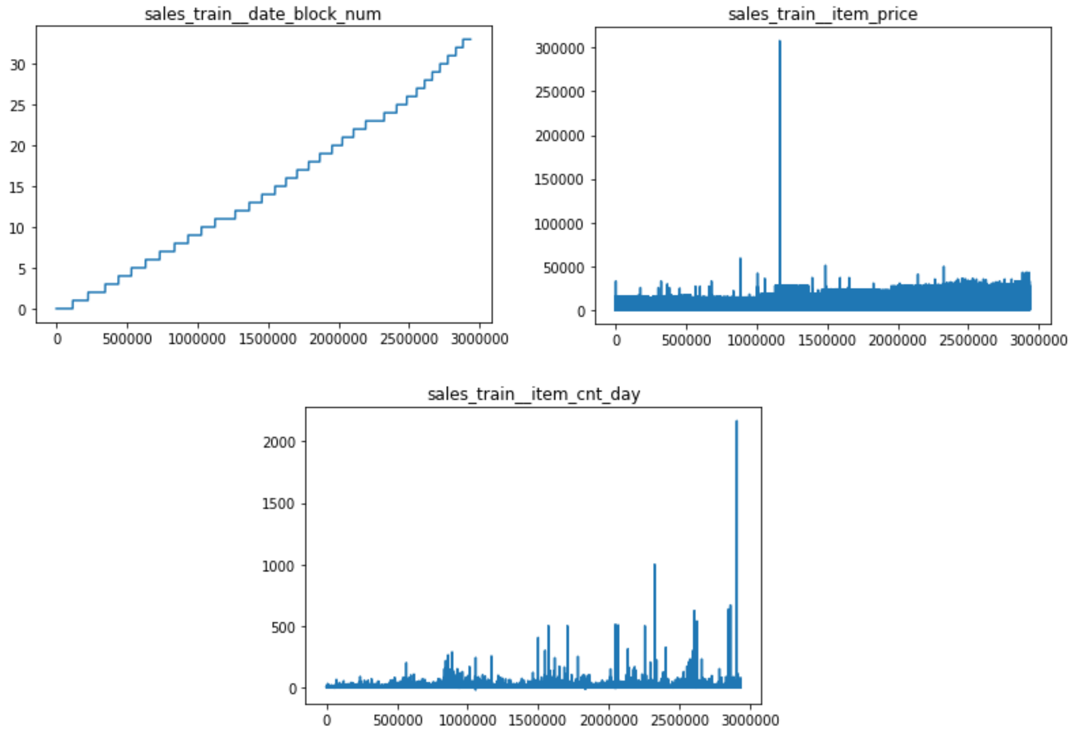

Figure 3.1: Anomaly detection

3.2. Preparation of data for machine learning

1. Adding information about city (*city_id*) and regrouping categories IDs (adding 18 main categories instead of original 84)

2. Summing month sales for every shop and choosing data from shops and items located in the test set. Full dataset consists of **7497000** vectors (214200 data from test set * 35 months)

3. Adding new features:

    - sales of item in a given shop in previous months

    - average sales of item in a given city in previus months

- average sales of item in every shop in previous months
- average sales of items belonging to the same category in a given shop in previous months
- average sales of items belonging to the same category in a given city in previous months
- average sales of items belonging to the same category in a every shop in previous months

4. Splitting the data into training, validation and test set:

    - **training set**: all data between the 12th and the 32nd month (data from 0-11th month are not considered in training because new features are generated for maximum of 12 previous months and for the 11th month there is no full information about sales in that period of time - 7068600 vectors
    - **validation set**: all data from the 33rd month (October 2015); validation set is used for early stop; validation error needs to decrease at least every 10 rounds to continue the training - 214200 vectors
    - **test set**: all data from 34 months (November 2015) - 214200 vectors

4. Machine Learning algorithms

4.1. XGBoost model

Table 4.1: XGBoost model parameters

Parameter	Value
Maximum tree depth	10
Number of boosted trees to fit	1000
Maximum rounds number	1000
Minimum sum of weight needed in a child	300
Subsample ratio of the training instance	0.8
Subsample ratio of columns when constructing each tree	0.8
Learning rate	0.3

4.2. Neural network

Table 4.2: Neural network architecture

Layer type	Number of neurons	Activation function
fully connected	256	ReLU
fully connected	128	ReLU
fully connected	64	ReLU
fully connected	32	ReLU
fully connected	1	-

Table 4.3: Neural network parameters

Parameter	**Value**
Batch size	512
Maximum number of epochs	25
Learning rate	0.001
Cost function	RMSE
Optimialization algorithm	Adam
Dropout rate	1.0

5. Presentation of results

5.1. Datasets

- Dataset_1 - 10 features : month, shop ID, city ID, item ID, category ID, item sales in a given shop in previous months (5 features).

- Dataset_2 - 20 features : month, shop ID, city ID, item ID, category ID, item sales in a given shop in previous months (5 features), average item sales in a given city in previous months (5 features), average item sales in previous months (5 features)

- Dataset_3 - 25 features : month, shop ID, city ID, item ID, category ID, item sales in a given shop in previous months (5 features), average item sales in a given city in previous months (5 features), average item sales in previous months (5 features), average sales of items belonging to the same category in a given shop in previous months (5 features).

Sales data from given month take into consideration the sales data obtained in the 1st, 2nd, 3rd, 6th and 12th month previous to this given month.

5.2. RMSE values

Table 5.1: The results of tests

Dataset	Algorithm	RMSE training data	RMSE validation data	RMSE test data
Baseline				1,168
Dataset_1	XGBoost	0.785	0.882	1.014
	Neural network	0.790	0.892	1.026
Dataset_2	XGBoost	0.776	0.879	1.008
	Neural network	0.781	0.888	1.024
Dataset_3	XGBoost	0.783	0.878	**1.006**
	Neural network	0.783	0.888	1.022

Test set consists of 35% of data from full test set (214200 vectors).

5.3. Feature importances

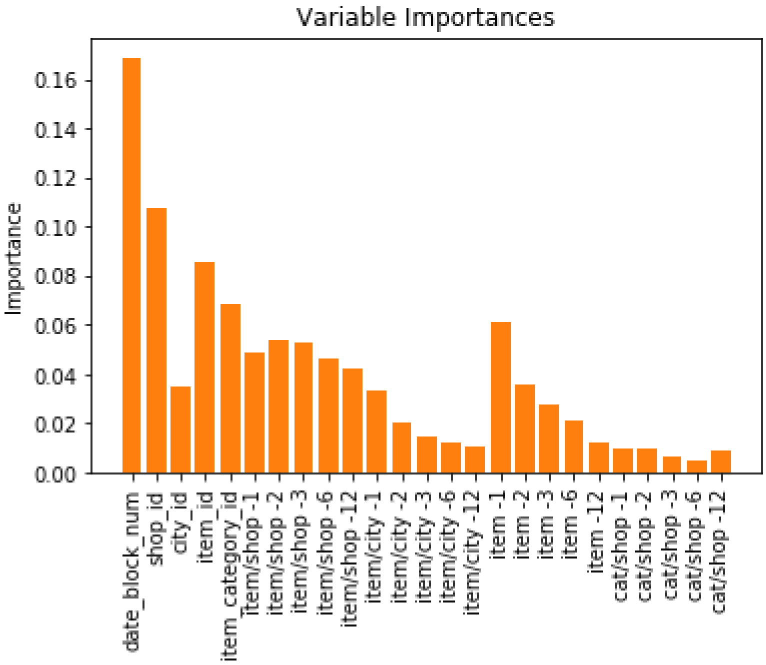

Figure 5.1: Feature importance in XGBoost model.

5.4. Neural network training

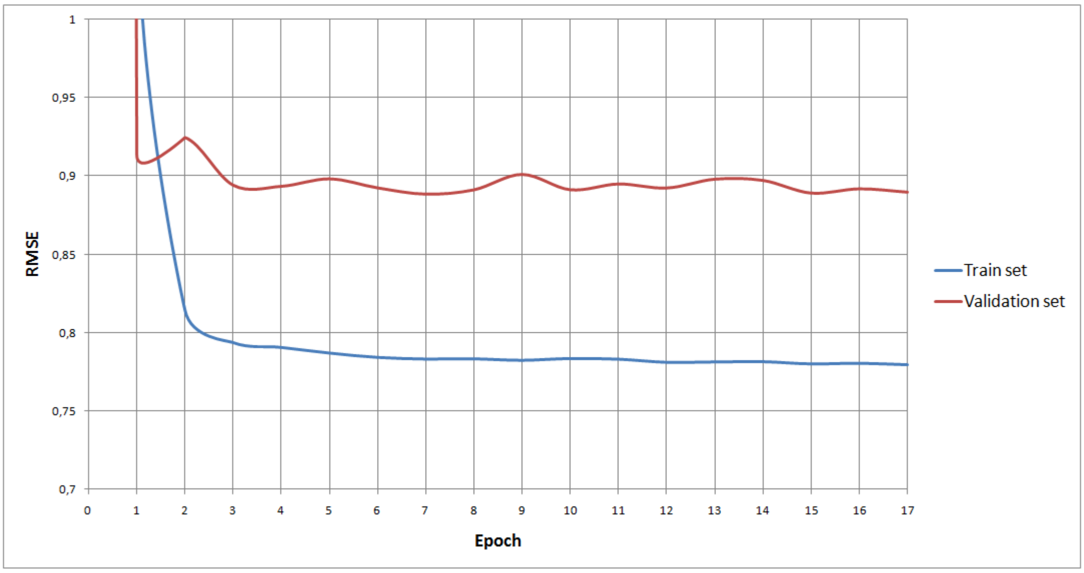

Figure 5.2: RMSE values for train set and validation set in every epoch

6. Conclusions

- The RMSE value was reached at 1.006, which means an improvement compared to the baseline by 0.162.
- Obtained results allowed to achieve 775th place out of 2055 projects taking part in *Kaggle Cometition* [as at 06/01/2019].
- Additionally, the efficiency of multi-layer neural network was tested - the results achieved are similar to those reached using the XGBoost model (the RMSE value at 1.022 would allow to achieve 842rd place).
- Limiting the data to only shop/item pairs occurring in test set allow to increase the speed of model training and reduce the computational complexity.
- Furthermore, the information about the city where the shop is located is used as feature in training.

6.1. Possible ways to develop the project

- Increasing the number of data used in training. First idea is to create, for every month, a grid from all shops/items combinations from that month (10913850 input vectors). Second idea is to use a combination of all stores and all products in each month (22170 products * 57 stores * 35 months = 44229150 vectors).

- Adding new features, such as, for example: overall sales of items belonging to the same category, item price or number of month from last sale of product in a given shop.

- Unfortunately, due to limited computational capabilities, it was impossible to carry out additional tests.