Final project: predict future sales

Team member
Fangshuo LIU
Liying FANG
Yujia DING

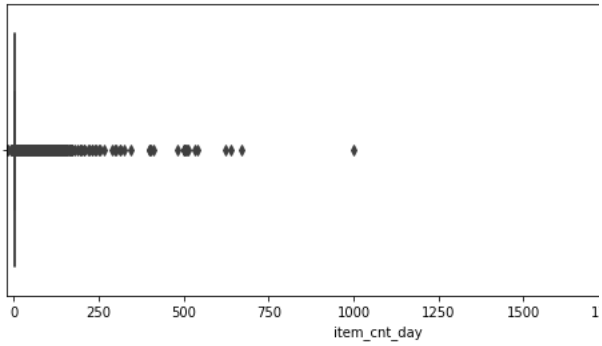
1. Topic Outline

In the real business world, it's of great importance to be forward-looking and be flexible. For retailer giants like 1C company in Russia, they pay quite a lot attention on predicting based on their considerable product and customer data and make continuous adjustment accordingly. In this analysis, with the help of 1C company, we try to dive deep into its sales history in the past few years and try to predict the expected sales for over 200k items in all 60 shops within Russia in this November. This analysis is expected to help 1C company better prepare for the incoming promising Thanksgiving sale festival, including but not limited to pricing, inventory management and so on.

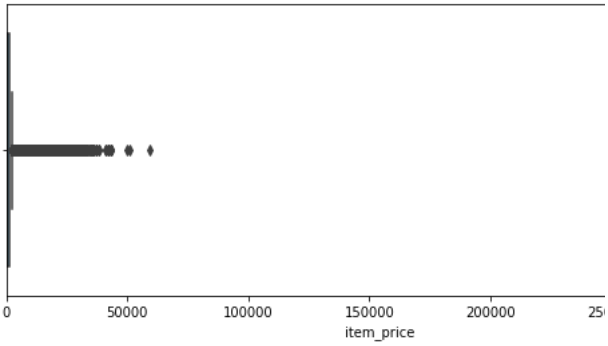
2. Data Introduction and Preprocessing

To make the prediction, we first need to conduct some exploratory analysis and conduct some basic data cleaning for further processing. Given the data ranging from shop info *("shops")*, item info *("items", "item_categories")*, previous detailed sales dataset for training *("sales_train")* and item list for testing prediction *("test")*, we'll mainly make use of the latter two, with nearly three million and over 200 thousand data separately, in this analysis. Note that although the given item info is described in Russian, this will not affect our results.

To establish our prediction model, we first need to check and clean the sales history data. In this data, for each row, we recorded the daily sales for each item in each shop in each day under certain price. Similar to reality, the list of items and shops will slightly varies across different months. With the help of boxplots, we excluded a few abnormal records with daily sales volume over 500 as well as those with sales price higher than 50000 rubles. Apart from outliers, we also checked and drop a few duplicates.



Item daily sales

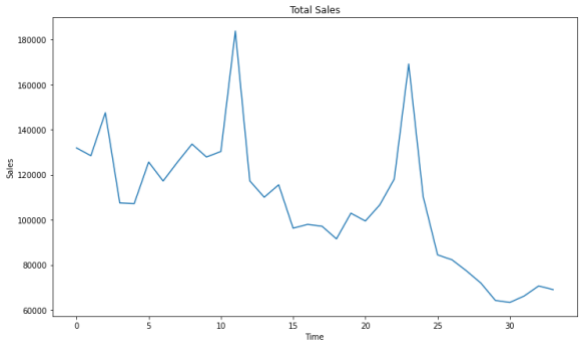


Item price

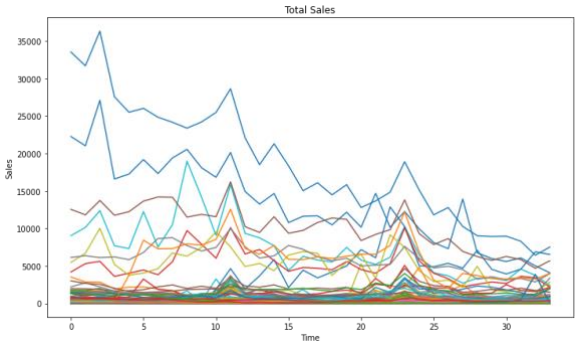
There are still a few abnormal values! For example, some weird empty or negative records occurs in daily sales, which may be a result of return of goods and should not be deleted as we mainly focus on the initial sales volume. We simply replace these records with 0. After implementing the

sales data with the item category info, we also observe some missing value in item price and fill them with median price of items in the same category. For datetime data, we strictly limit the dataset within the range from Jan 1, 2013 to Oct 31, 2015 (labelling each month from 0 to 33). Until now, we've finished the first-step cleaning of our data.

As you may see from our previous introduction, the time factor may play an important role in this time-series dataset. Therefore, we conduct a seasonality and trending analysis as below. Either the total sales or the sales by category could easily prove our judgment that there is an obvious "seasonality" and a decreasing "trending" for our existing items. Specifically, within each year, we observe a sales peak in the winter, which is probably out of the yearly long-lasting sales festival from November. Meanwhile, from 2013 to 2015, the overall sales is frustratedly reducing, and that's just the opposite to our expectation, and some adjustments on our pricing and listing strategy based on our prediction results is therefore strongly required. We then reformat our dataset through replacing the single season feature with separate season dummies (fall is excluded given the potential collinearity) and aggregating monthly sales per item to prepare for the prediction of monthly sales.



Total sales



Sales per category

3. Modeling

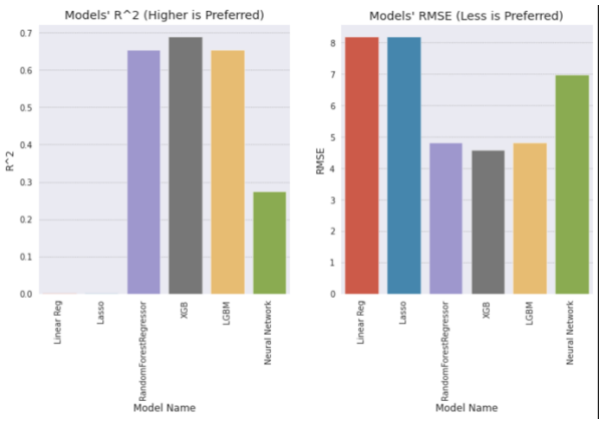
For better interpretating our following models like neural network, we scaled our sales history dataset and split them into train and test data with a 75%:25% composition. We'll try with a series of predicting models to figure out the optimal ones, and use it to finally predict the monthly sales in the next month. Here for each model, we provide a brief explanation for your reference.

- *Linear regression, LASSO*: after we conduct the most basic model, the result indicates that linear regression is absolutely far from the optimal model for this dataset. It's hard to imagine that an optimal model can only result in a R^2 at the level of 0.001. Aiming to step a bit forward, we then tried the regularized LASSO for feature selection. However, just as we can conclude from the close to zero improvement for LASSO, given we only have a few number of features, LASSO is not a proper approach as well.
- *Random forest*: within this model, we will establish our prediction taking the feature importance into consideration. For improving the prediction accuracy, random forest has always proved to preform pretty well in our train dataset, with a jump in R^2 and obvious reduction in RMSE compared with linear regression. Here we gradually increases the max depth from 5 to 20, for which a smaller value indicates a weaker estimator, to achieve a better prediction result.

- *LGBM*: apart from random forest, we also consider other models for better accuracy. LGBM, also known as LightGBM, provides one approach in the view of only including those data with significant gradients for info gain estimation. Compared with linear regression, it seems LGBM is at the advantage. Although the overall accuracy is a bit weaker than random forest, LGBM performs better in its computing speed.
- *XGBoost*: we also introduce another gradient boosting named XGBoost as a comparison. As one of the most popular algorithms, XGBoost achieve a comparatively most satisfying estimation result.
- *Neural Network*: last but not least, considering the complexity of our dataset, we tried neural network with the ReLU as activation function as well. Given the range of our output sales volume is theoretically zero to infinity, ReLU is the optimal choice compared with other activation functions like Sigmoid. However, the modeling results is under our expectation and thus is not chosen as the final algorithm.

To sum up, the comparison of RMSE and R^2 of all models is listed as below.

Model name	RMSE	R^2
Linear	8.186544	0.001335
LASSO	8.186546	0.001335
Random forest	4.814671	0.654577
LGBM	4.817174	0.654217
XGBoost	4.564902	0.689486
Neural network	6.973661	0.275330



As lower level of RMSE and higher level of R^2 is preferred for an optimal model, we finally choose **XGBoost** as the target model on our test data.

4. Prediction Results

Now we have confirmed our model for prediction. Before we start our testing, we first need to reformat the test dataset to align with training data. Specifically, we implement the item category, sales month, and season dummies, and filling the item price based on the price data extracted from training dataset aggregated on the item category level. As our next month (November) belongs to winter, our referred price data is only from the sales history data in the previous winter to be more precise. Our adjusted test dataset is as below.

	date_block_num	shop_id	item_id	item_category_id	item_price	spring	summer	winter
0	34	2	30	40.0	149.0	0	0	1
1	34	2	31	37.0	399.0	0	0	1
2	34	2	32	40.0	149.0	0	0	1
3	34	2	33	37.0	199.0	0	0	1
4	34	2	38	41.0	699.0	0	0	1
...	...	...	...	...	...	...	...	...
214195	34	59	22162	40.0	149.0	0	0	1
214196	34	59	22163	40.0	149.0	0	0	1
214197	34	59	22164	37.0	299.0	0	0	1
214198	34	59	22166	54.0	150.0	0	0	1
214199	34	59	22167	49.0	299.0	0	0	1

214200 rows × 8 columns

Test dataset after reformatting

We then fit the model with our testing data. Here we provide a preview for our resulting sales prediction for your reference. Note that the *ID* column can be seen as combination of *shop id* and *item id* and thus be unique for certain item in different stores.

	ID	item_cnt_month
0	0	0.000000
1	1	1.731981
2	2	0.000000
3	3	0.000000
4	4	6.322827
...	...	...
214195	214195	9.730357
214196	214196	9.730357
214197	214197	2.119448
214198	214198	0.116273
214199	214199	0.963130

214200 rows × 2 columns

Output sales prediction

Based on the existing sales history and this prediction analysis, 1C company could make proper planning in the next few weeks for the incoming Thanksgiving sale to better match the market demand with its inventory and maximize the net revenue. Meanwhile, 1C company could further consider to optimize its item list distribution in different shops including removing those out-of-date items with trending items to relieve itself from the current trap since the past few years.

Competition: https://www.kaggle.com/c/competitive-data-science-predict-future-sales/overview