Rossmann store sales quantity prediction

Sazontyev V.V., Stanford University, mentor: Derek Lim

Motivation

Rossmann store managers are tasked with predicting their daily sales for up to six weeks in advance. Store sales are influenced by many factors.

Goal of the project

The basic idea is to build a good machine learning model. And my personal goal is to show, that if we analyze learning curve we can achieve really good model. (based on analysis, not based on gut feelling, or popularity of the model among Kaggle contestants).

Data visualization and analysis

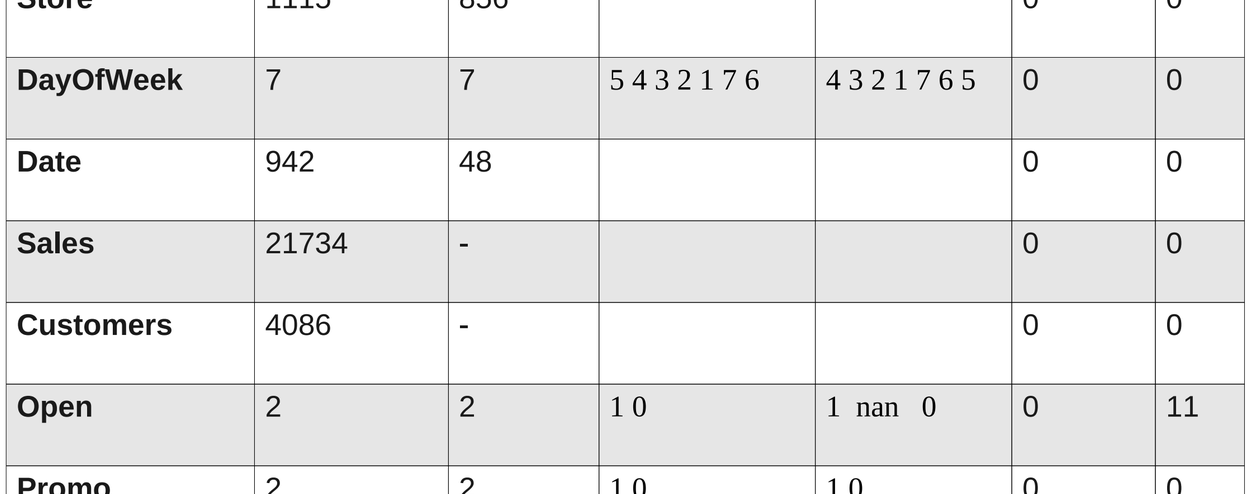
I start basically with outputting all columns by files of given dataset.

Train	Store	Test
Store	Store	Id
DayOfWeek	StoreType	Store
Sales	Assortment	DayOfWeek
Customers	CompetitionDistance	Date
Open	CompetitionOpenSinceMonth	Open
Promo	Promo2	Promo
StateHoliday	Promo2SinceWeek	StateHoliday
SchoolHoliday	Promo2SinceYear	SchoolHoliday
	PromoInternal	

The first, that we see, that columns are named, and have meaningful names and descriptions.
The next table shows amount of unique values, nans, and unique values if there is less than 10.

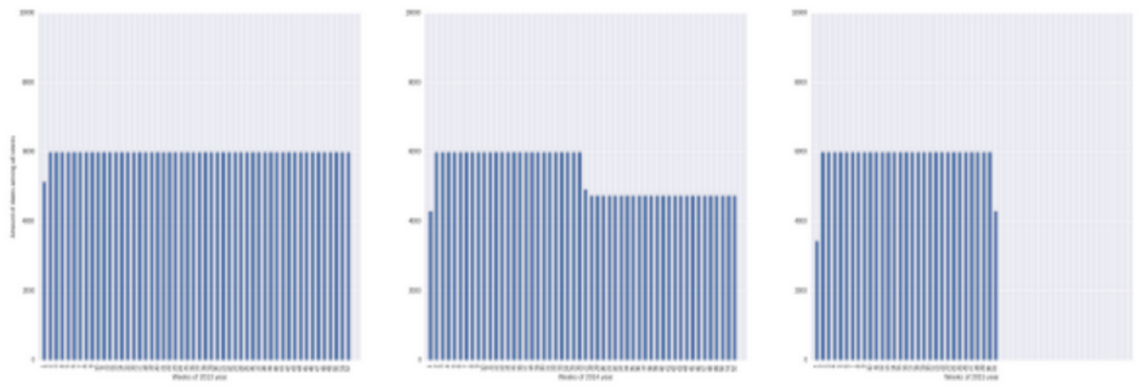
	Amount of unique values		Unique values		NaNs	
Field name	**Training set**	**Test set**	**Training set**	**Test set**	**Training**	**Test**
Store	1115	856			0	0
DayOfWeek	7	7	5 4 3 2 1 7 6	4 3 2 1 7 6 5	0	0
Date	942	48			0	0
Sales	21734	-			0	0
Customers	4086	-			0	0
Open	2	2	1 0	1 nan 0	0	11
Promo	2	2	1 0	1 0	0	0
StateHoliday	5	2	'0' 'a' 'b' 'c' 0L	'0' 'a'	0	0
SchoolHoliday	2	2	1 0	1 0	0	0

From this table we see, that there are 11 NaNs in “Open” on test. Also in test set, there only 0,'a' value in StateHoliday field(it's different from what we see in training set). On the next picture we see, that



distribution among all sales (left pic). I took a logarithm of it, and it become even more bell-shaped (right pic).

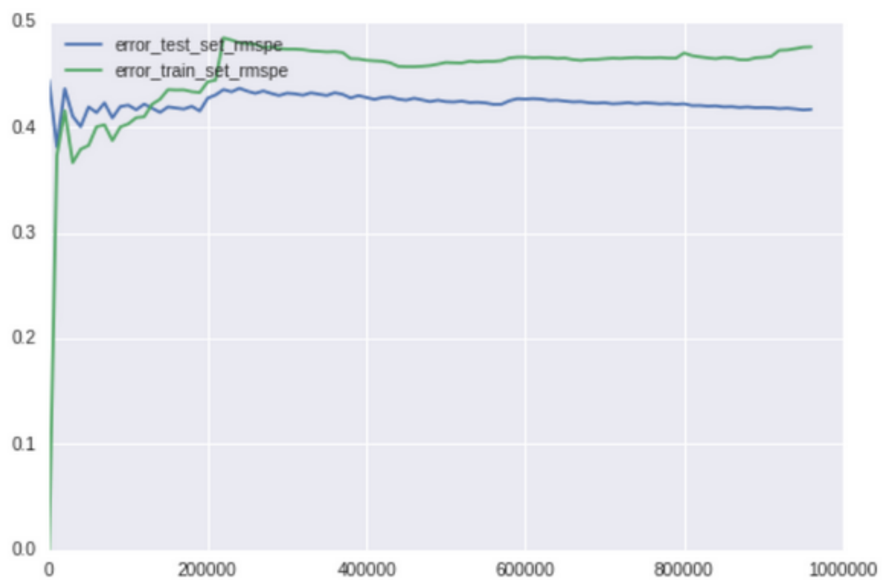
I plotted graph date days in training set amount of days by weeks and by years among all stores. From this plot, we see, that



I found, that statistics for 180 store ids are missing since 27-th week up to 52 week. But they present in test set (we should predict for them).
One more point, that I found, that last 5 days of month greatly changes the behavior of sales curve.

Evolution of machine learning models.

I started, with most simple model – LinearRegression (using sklearn.linear_model.LinearRegression()). This model is a baseline. I used it without any regularization, since we clearly do not overfit out data. As features I used: Store and DayOfWeek I got 0.4761 on train set and 0.4170 on test set (based on RMPSE) on local cv evaluation.



Training and test set error. x-axis - amount of training examples; y-axis RMSPE error.

Learning curve for baseline algorithm.

Based on figure, next step that I made was to take mean of sales grouped by Store and DayOfWeek, and report that mean as a prediction that gave me on my local cv evaluation - 0.1567 (RMSPE). The intuition behind it, that we made more complex model, with same features. When I added Promo and Open feature in this model my model showed me on my local cv evaluation 0.14121 and on public leader board 0.13693. Also, as Derek Lim commented – I deal with time-series, I added weights based on week number (more recent week – more valuable it is). I tried to used these weighting schemes:

$$w_i^N = \left(\frac{d-i+1}{d}\right)^{\delta} \quad \delta \in [0,+\infty) \qquad w_i^N = \lambda^i \quad \lambda \in (0,1] \qquad w_i^N = \frac{1}{i^{\gamma}}, \quad \gamma \in [0,+\infty)$$

Unfortunately, no one of them did not worked.
Next model what I implemented is locally constant model with linear regression grouped by same features, in other words – now I report not just mean – I train linear regression for each split. It does improved my model by 0.022 on my local cv and 0.02 on public leader board. Then since this model is still doesn't as complex as should be, I moved to xgboost – boosted (tree) algorithms. It gave me 0.11520 on public leader board.