Corporación Favorita Grocery Sales Forecasting

Eric Chen

The Kaggle Competition

https://www.kaggle.com/c/favorita-grocery-sales-forecasting

The Challenge: grocery stores are always in a delicate dance with purchasing and sales forecasting. Predict a little over, and grocers are stuck with overstocked, perishable goods. Guess a little under, and popular items quickly sell out, leaving money on the table and customers fuming. The problem becomes more complex as retailers add new locations with unique needs, new products, ever transitioning seasonal tastes, and unpredictable product marketing.

The Goal: To build a model that more accurately forecasts product sales.

Evaluation

Submissions are evaluated on the Normalized Weighted Root Mean Squared Logarithmic Error (NWRMSLE), calculated as follows:

$$NWRMSLE = \sqrt{\frac{\sum_{i=1}^{n} w_i (\ln(\hat{y}_i + 1) - \ln(y_i + 1))^2}{\sum_{i=1}^{n} w_i}}$$

This metric is suitable when predicting values across a large range of orders of magnitudes. It avoids penalizing large differences in prediction when both the predicted and the true number are large: predicting 5 when the true value is 50 is penalized more than predicting 500 when the true value is 545.

Data

holidays_events.csv: Holidays and Events, with metadata. 351 rows

items.csv: Item metadata, including family, class, and perishable. 4101 rows.

oil.csv: Daily oil price. Includes values during both the train and test data timeframe. (Ecuador is an oil-dependent country and it's economical health is highly vulnerable to shocks in oil prices.) 1219 rows.

stores.csv: Store metadata, including city, state, type, and cluster. 55 rows.

test.csv: Test data, with the date, store_nbr, item_nbr combinations that are to be predicted, along with the onpromotion information. 3,000,000 rows

train.csv: Training data, which includes the target unit_sales by date, store_nbr, and item_nbr and a unique id to label rows. 125,000,000 rows

transactions.csv: The count of sales transactions for each date, store_nbr combination. 83489 rows.

Data cleaning

merge item, store, oil

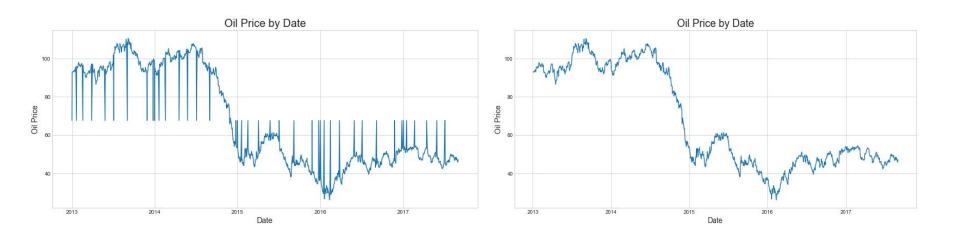
problem - missing oil data.



Visualization is important

oil price with fillna mean

oil price with fillna ffill



Feature engineering

mean_sales

payday

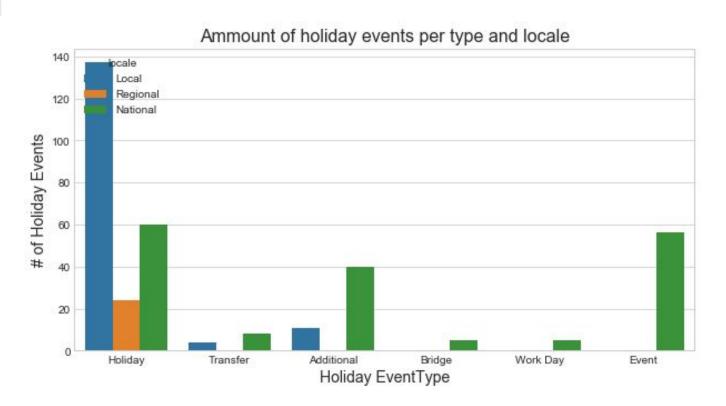
weekday

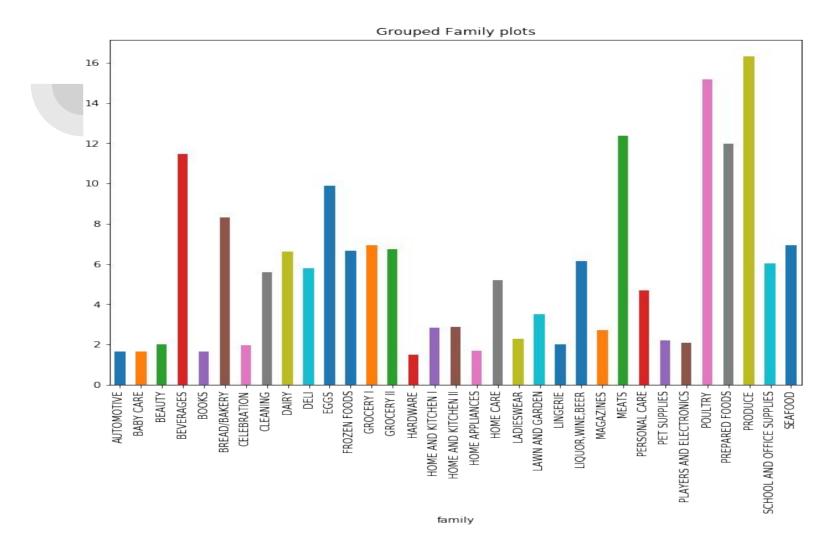
day

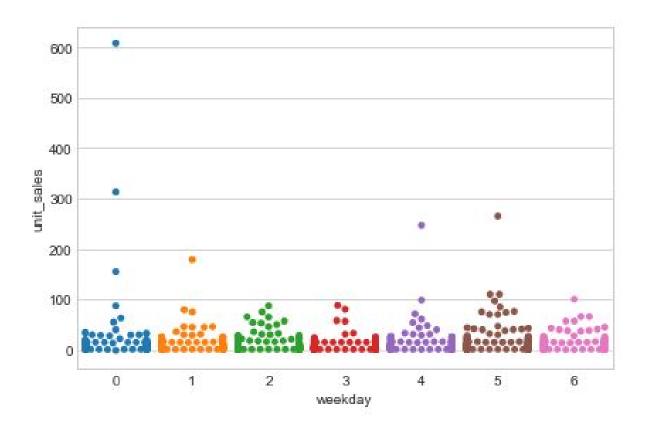
month

dayofyear

Exploratory data analysis







- 0.75

- 0.50

- 0.25

- 0.00

- -0.25

store_nbr	1	0.0087	0.045	-0.0059	0.0079	0.0044	0.0053	0.0031	0.017	0.017	0.058	0.00069	-0.0071	0.023
item_nbr	0.0087	1	0.024	0.019	0.035	0.044	0.0027	0.00023	0.015	0.015	0.031	0.00061	-0.0053	0.0012
unit_sales	0.045	0.024	1	0.084	-0.051	0.06	0.028	-0.013	-0.0018	-0.0035	0.77	0.00017	0.0066	0.022
onpromotion -	-0.0059	0.019	0.084	1	0.048	0.12	-0.013	0.0088	0.013	0.014	0.11	0.0042	-0.017	-0.0078
dass	0.0079	0.035	-0.051	0.048	1	0.16	0.0082	-0.0031	-0.0021	-0.0025	-0.066	0.00063	0.0021-	0.00045
perishable ·	0.0044	0.044	0.06	0.12	0.16	1	-0.0064	0.0033-	0.00033	0.00012	0.078	0.00044	10.0026	-0.0045
weekday ·	0.0053	0.0027	0.028	-0.013	0.0082	-0.0064	1	0.0084	0.011	0.012	0.036	-0.025	-0.087	-0.0016
day -	0.0031	0.00023	-0.013	0.0088	-0.0031	0.0033	0.0084	1	-0.097	0.036	0.0049	0.22	-0.094	6.7e-05
month :	0.017	0.015	-0.0018	0.013	-0.0021-	0.00033	0.011	-0.097	1	0.99	0.0046	-0.0059	-0.38	0.0053
dayofyear	0.017	0.015	-0.0035	0.014	-0.0025	0.00012	0.012	0.036	0.99	1	0.0052	0.023	-0.4	0.0054
mean_sales	0.058	0.031	0.77	0.11	-0.066	0.078	0.036	0.0049	0.0046	0.0052	1	0.00035	0.0012	0.028
is_payday ·	0.00069	0.00061	0.00017	0.0042	0.00063	0.00044	-0.025	0.22	-0.0059	0.023	0.0003	1	-0.034	0.00034
dcoilwtico	-0.0071	-0.0053	0.0066	-0.017	0.0021	-0.0026	-0.087	-0.094	-0.38	-0.4	-0.0012	-0.034	1	-0.0026
duster	0.023	0.0012	0.022	-0.0078	0.00045	0.0045	-0.0016	6.7e-05	0.0053	0.0054	0.028	0.00034	10.0026	1
	store_nbr -	item_nbr -	unit_sales -	anpromotion -	dass -	perishable -	weekday	- day	month -	dayofyear -	mean_sales -	is_payday -	dcoilwtico -	duster -

Data transformation

- Convert categorical data using one hot encoding then truncatedSVD to reduce dimension
 - family, city, state, type, weekday, day, month, cluster
- Scale mean_sales and oil price (dcoilwtico)
- item_id too large for one hot encoding, use mean encoding instead.

		A		
X0	x1	x2	xn	У
0.17	0.25	0.93	0.79	1
0.35	0.61	0.93	0.57	0
0.44	0.59	0.56	0.46	0
0.37	0.43	0.74	0.28	1
0.96	0.07	0.57	0.01	1

		В		
X0	x1	x2	xn	У
0.89	0.72	0.50	0.66	0
0.58	0.71	0.92	0.27	1
0.10	0.35	0.27	0.37	0
0.47	0.68	0.30	0.98	0
0.39	0.53	0.59	0.18	1

c							
X0	x1	x2	xn	У			
0.29	0.77	0.05	0.09	?			
0.38	0.66	0.42	0.91	?			
0.72	0.66	0.92	0.11	?			
0.70	0.37	0.91	0.17	?			
0.59	0.98	0.93	0.65	?			

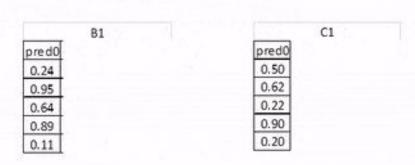
Consider datasets A,B,C. Target variable (y) is known for A,B

		Α		
X0	x1	x2	xn	У
0.17	0.25	0.93	0.79	1
0.35	0.61	0.93	0.57	0
0.44	0.59	0.56	0.46	0
0.37	0.43	0.74	0.28	1
0.96	0.07	0.57	0.01	1

		В		
X0	x1	x2	xn	У
0.89	0.72	0.50	0.66	0
0.58	0.71	0.92	0.27	1
0.10	0.35	0.27	0.37	0
0.47	0.68	0.30	0.98	0
0.39	0.53	0.59	0.18	1

С						
X0	x1	x2	xn	У		
0.29	0.77	0.05	0.09	?		
0.38	0.66	0.42	0.91	?		
0.72	0.66	0.92	0.11	?		
0.70	0.37	0.91	0.17	?		
0.59	0.98	0.93	0.65	?		

Train algorithm 0 on A and make predictions for B and C and save to B1, C1



		Α		
X0	x1	x2	xn	У
0.17	0.25	0.93	0.79	1
0.35	0.61	0.93	0.57	0
0.44	0.59	0.56	0.46	0
0.37	0.43	0.74	0.28	1
0.96	0.07	0.57	0.01	1

		В		
X0	x1	x2	xn	У
0.89	0.72	0.50	0.66	0
0.58	0.71	0.92	0.27	1
0.10	0.35	0.27	0.37	0
0.47	0.68	0.30	0.98	0
0.39	0.53	0.59	0.18	1

C							
XO	x1	x2	xn	У			
0.29	0.77	0.05	0.09	?			
0.38	0.66	0.42	0.91	?			
0.72	0.66	0.92	0.11	?			
0.70	0.37	0.91	0.17	?			
0.59	0.98	0.93	0.65	?			

Train algorithm 0 on A and make predictions for B and C and save to B1, C1
Train algorithm 1 on A and make predictions for B and C and save to B1, C1
Train algorithm 2 on A and make predictions for B and C and save to B1, C1

B1						
pred0	pred1	pred2	У			
0.24	0.72	0.70	0			
0.95	0.25	0.22	1			
0.64	0.80	0.96	0			
0.89	0.58	0.52	0			
0.11	0.20	0.93	1			

C1						
pred0	pred1	pred2	у			
0.50	0.50	0.39	?			
0.62	0.59	0.46	?			
0.22	0.31	0.54	?			
0.90	0.47	0.09	?			
0.20	0.09	0.61	?			

		Α		
XO	x1	x2	xn	У
0.17	0.25	0.93	0.79	1
0.35	0.61	0.93	0.57	0
0.44	0.59	0.56	0.46	0
0.37	0.43	0.74	0.28	1
0.96	0.07	0.57	0.01	1

		В		
XO	x1	x2	xn	У
0.89	0.72	0.50	0.66	0
0.58	0.71	0.92	0.27	1
0.10	0.35	0.27	0.37	0
0.47	0.68	0.30	0.98	0
0.39	0.53	0.59	0.18	1

reast .		C		
X0	×1	x2	xn:	У
0.29	0.77	0.05	0.09	?
0.38	0.66	0.42	0.91	?
0.72	0.66	0.92	0.11	?
0.70	0.37	0.91	0.17	?
0.59	0.98	0.93	0.65	?

Train algorithm 0 on A and make predictions for B and C and save to B1, C1
Train algorithm 1 on A and make predictions for B and C and save to B1, C1
Train algorithm 2 on A and make predictions for B and C and save to B1, C1

B1			
pre d0	pred1	pred2	y.
0.24	0.72	0.70	0
0.95	0.25	0.22	1
0.64	0.80	0.96	0
0.89	0.58	0.52	0
0.11	0.20	0.93	1

C1				
pred0	pred1	pred2	У	
0.50	0.50	0.39	?	
0.62	0.59	0.46	?	
0.22	0.31	0.54	?	
0.90	0.47	0.09	?	
0.20	0.09	0.61	?	

0.45 0.23 0.99 0.34 0.05

Preds3

Train algorithm 3 on B1 and make predictions for C1

Conclusion

My model use XGBoost, Keras Regressor, Adaboast, RandomForestRegressor as first level of stacking with XGBoost on second level. resulting in .969, while the winning score is .509.

Looking at winning solutions feature engineering is very important.

Next step

Try algorithms from winning solutions,

Try time series algorithms