A LOOK INTO A COMPLEX TIME-SERIES DATASET

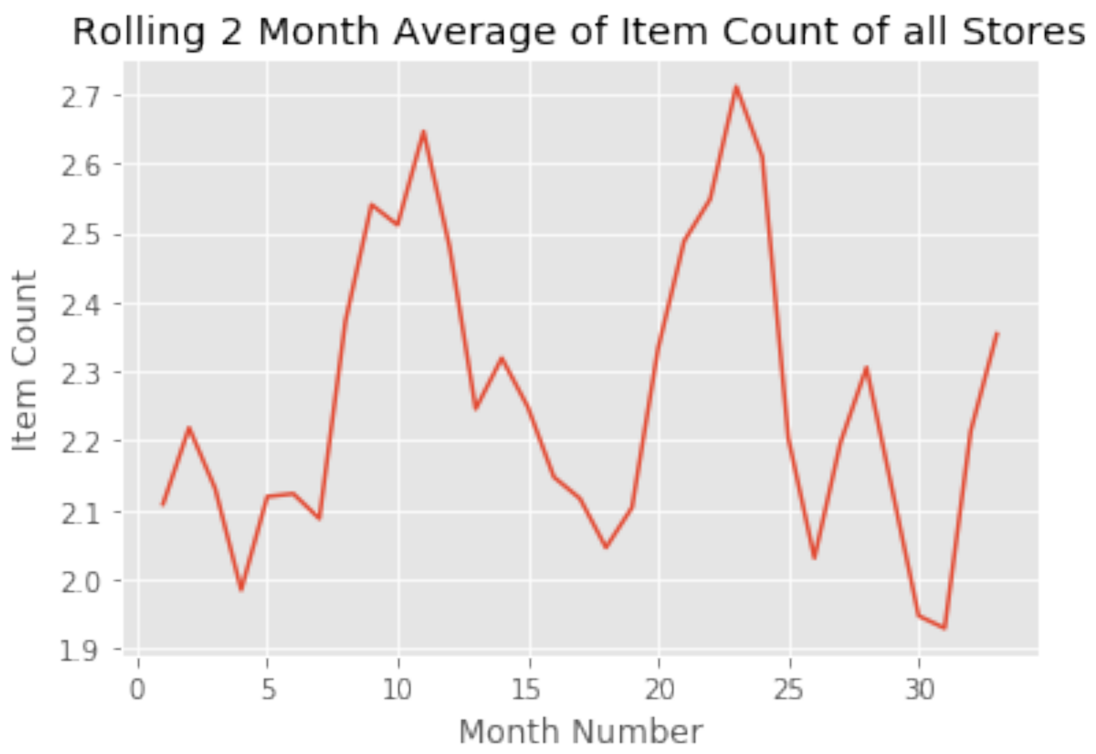
PREDICTING SALES VOLUME

DATA

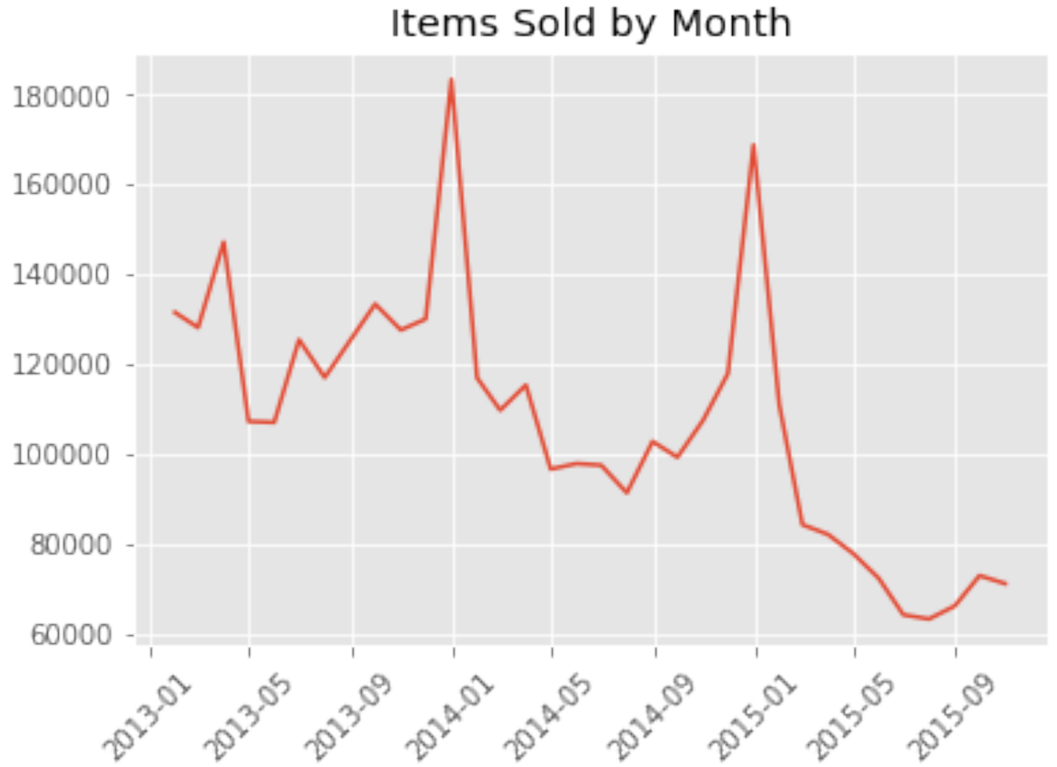
- Kaggle dataset
- From February 2013 to November 2015
- Almost 3 million rows of data
- Contains multiple shops, items, and item price
- Memory usage: 134.4+ MB

	date	date_block_num	shop_id	item_id	item_price	item_cnt_day
1	02.01.2013	0	59	22154	999.0	1.0
2	03.01.2013	0	25	2552	899.0	1.0
3	05.01.2013	0	25	2552	899.0	-1.0
4	06.01.2013	0	25	2554	1709.05	1.0
5	15.01.2013	0	25	2555	1099.0	1.0
6	10.01.2013	0	25	2564	349.0	1.0
7	02.01.2013	0	25	2565	549.0	1.0
8	04.01.2013	0	25	2572	239.0	1.0
9	11.01.2013	0	25	2572	299.0	1.0
10	03.01.2013	0	25	2573	299.0	3.0
11	03.01.2013	0	25	2574	399.0	2.0
12	05.01.2013	0	25	2574	399.0	1.0
13	07.01.2013	0	25	2574	399.0	1.0
14	08.01.2013	0	25	2574	399.0	2.0
15	10.01.2013	0	25	2574	399.0	1.0
16	11.01.2013	0	25	2574	399.0	2.0
17	13.01.2013	0	25	2574	399.0	1.0
18	16.01.2013	0	25	2574	399.0	1.0
19	26.01.2013	0	25	2574	399.0	1.0
20	27.01.2013	0	25	2574	399.0	1.0

EXPLORING THE DATA

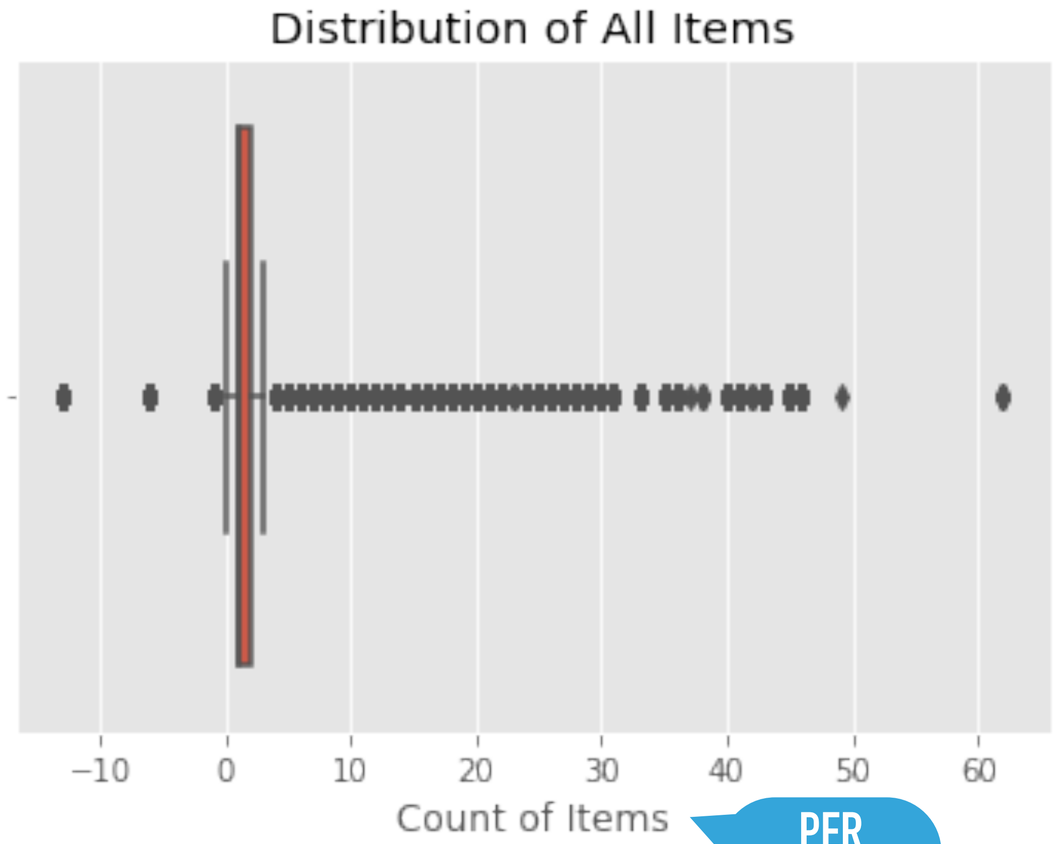


There is a seasonality trend that persists throughout the data. The two humps fall close to the end of the year at month 12 and 24.



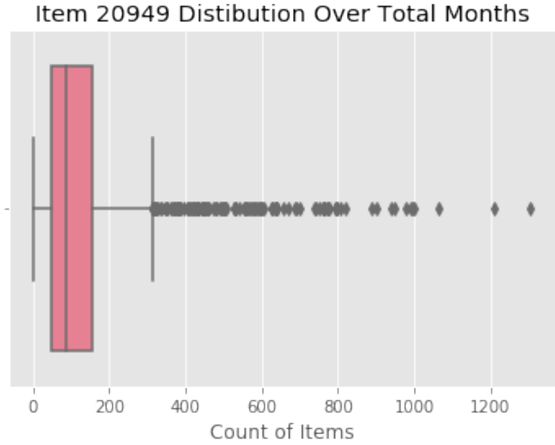
Amount is decreasing as the months go on.

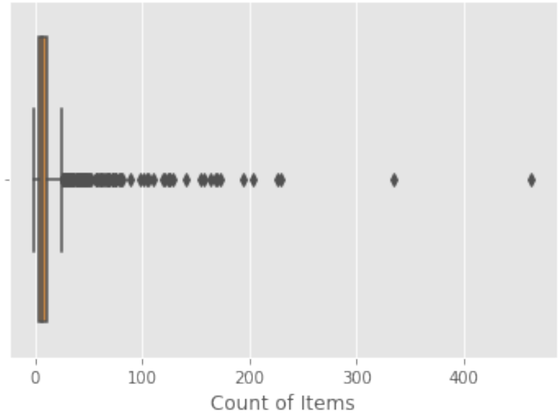
EXPLORING THE DATA

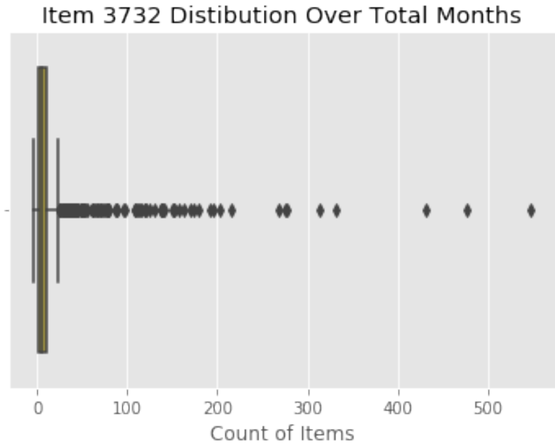


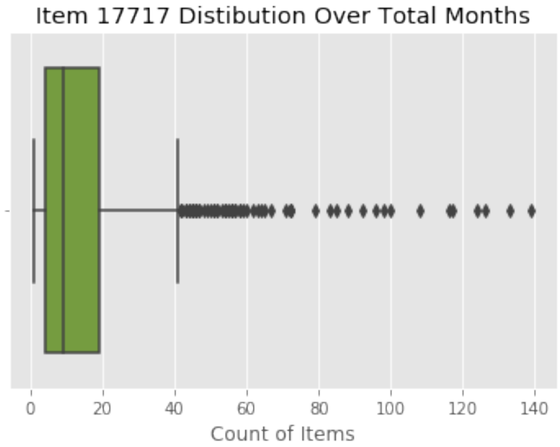
PER MONTH!

TOP 4 MOST SOLD ITEMS





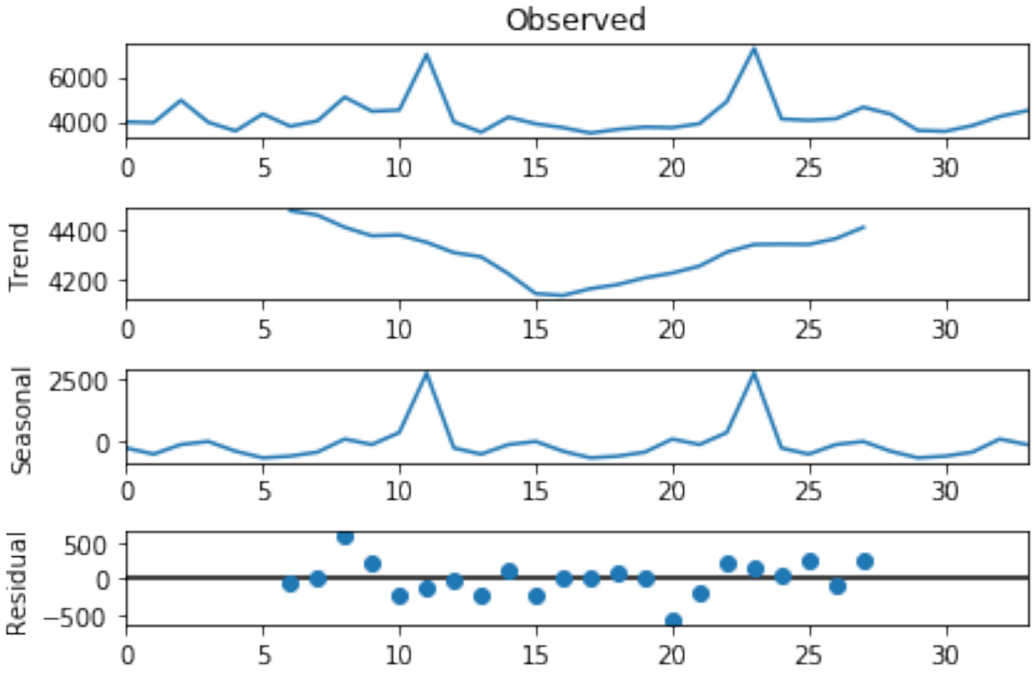




CONCLUSIONS FOUND IN EDA

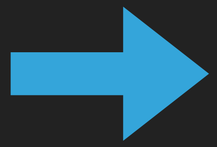
- Different items from month to month!!!
- Many items are close to zero but there is a wide range of outliers
- There is a seasonality in the data
- There is a trend in the data

SEASONAL DECOMPOSE FOR SHOP 42



TRANSFORMING DATA TO FIT MODEL

	date_block_num	item_id	item_cnt_month
0	0	28	0.0
1	0	30	0.0
2	0	31	0.0
3	0	32	11.0
4	0	33	2.0
...	...	...	...
409151	33	22162	0.0
409152	33	22163	1.0
409153	33	22164	3.0
409154	33	22167	4.0
409155	33	22168	0.0



date_block_num	0	1	2	3	4	5	6	7	8	9	...	24	25	26	27	28	29	30	31	32	33
item_id																					
28	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
30	0.0	46.0	22.0	7.0	3.0	3.0	3.0	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
31	0.0	39.0	12.0	3.0	2.0	1.0	2.0	2.0	1.0	1.0	...	3.0	2.0	1.0	1.0	0.0	5.0	1.0	7.0	0.0	2.0
32	11.0	5.0	1.0	6.0	6.0	6.0	3.0	2.0	3.0	1.0	...	0.0	0.0	0.0	1.0	1.0	2.0	0.0	0.0	0.0	2.0
33	2.0	1.0	2.0	1.0	0.0	5.0	1.0	0.0	2.0	2.0	...	2.0	4.0	3.0	1.0	0.0	1.0	2.0	1.0	0.0	1.0

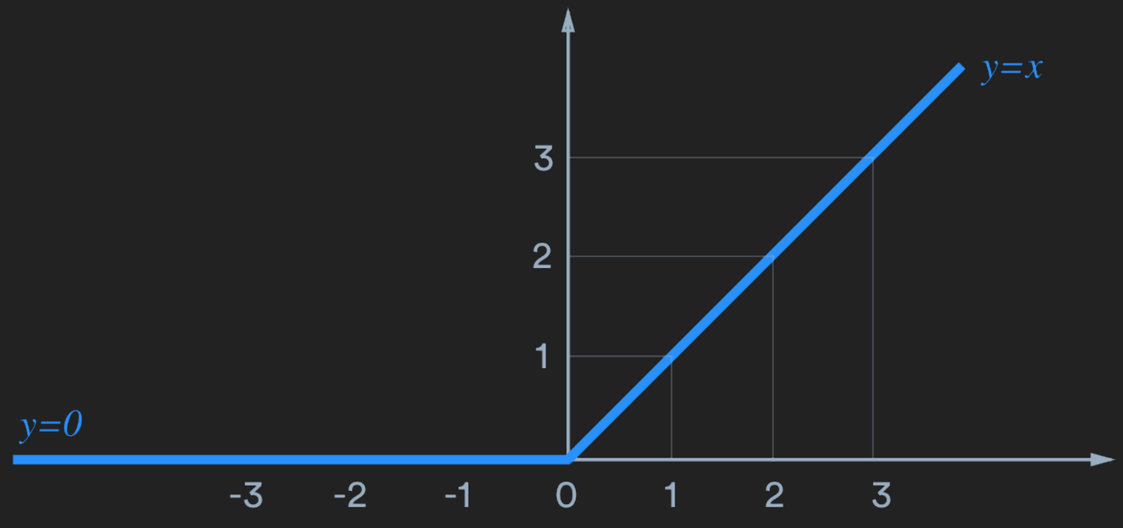
- Must add all unique items to every month as efficiently as possible. Set `int64` to `int16` for `date_block_num` and `item_id`.
- Fill in zeroes
- Pivot the data frame

TIME-SERIES LONG SHORT TERM MEMORY (LSTM)

- Dense layer with 64 nodes, `activation = 'relu'` (which does well for vanishing gradients)
- LSTM layer with 32 nodes, `activation = 'relu'`
- `EarlyStopping` with a patience of 5
- Epochs: 15
- Batch size: 512
- Learning rate: .0001
- Loss: `'mse'`

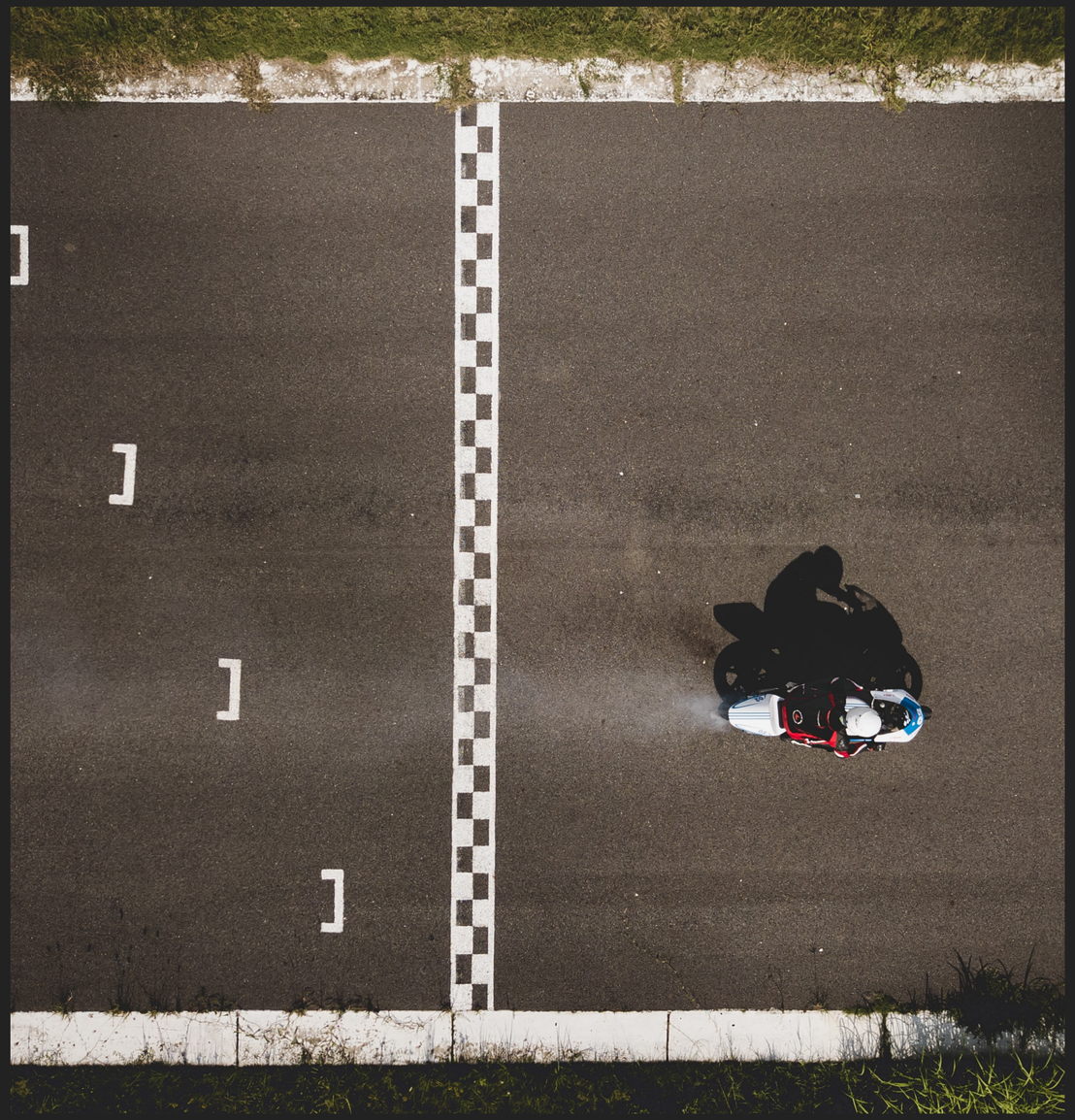
MOST VALIDATION LOSS:

< 1



TRANSFORMING TEST FOR SUBMISSION

- In order to make predictions, test file must be in the same shape as train file (do some transformations)
- Predict!
- Predictions were separated by shop
- Iterate through all shops (do some transformations)
- Concatenate all shops into one data frame
- Submit! (Finally!)



RMSE SCORE: 1.59829

Kaggle Score

CONCLUSIONS

- Preprocessing step is the most important step for time-series neural nets.
- Complex datasets → spend a lot of time in preprocessing
- It was interesting to see validation loss scores increase dramatically for specific stores – this may have something to do with outliers in that particular shop.
- Vast majority of items in test file should be 0. Can our model predict this?
- Understanding test file is important as well

NEXT STEPS

- Since RMSE is heavily affected by outliers, it would be best to remove any outliers.
- Remove stationarity
- I would like to try a multilayer perceptron (MLP) model which was mentioned in an article that compared LSTM's with MLP's.
- Using other gradient descent boosting techniques (XGBoost).
- There is an `item_category` column that could be useful to use in a model (making it a multi-variate model).