Predicting Future Sales

PANKAJ ACHARYA

SPRINGBOARD DATA SCIENCE CAREER TRACK

FEBRUARY 2019 COHORT

MENTOR: KEVIN GLYNN

The Problem

- To predict next month’s sales of items from different shops based on historical data to help optimize the stocks.

Why is this interesting?

- Having the product in stock for too long costs a lot.
- Short supply will also lead to lost of revenue and customer trust.
- In current situation of competition, maintaining a balance between over- and under- stock is challenging.

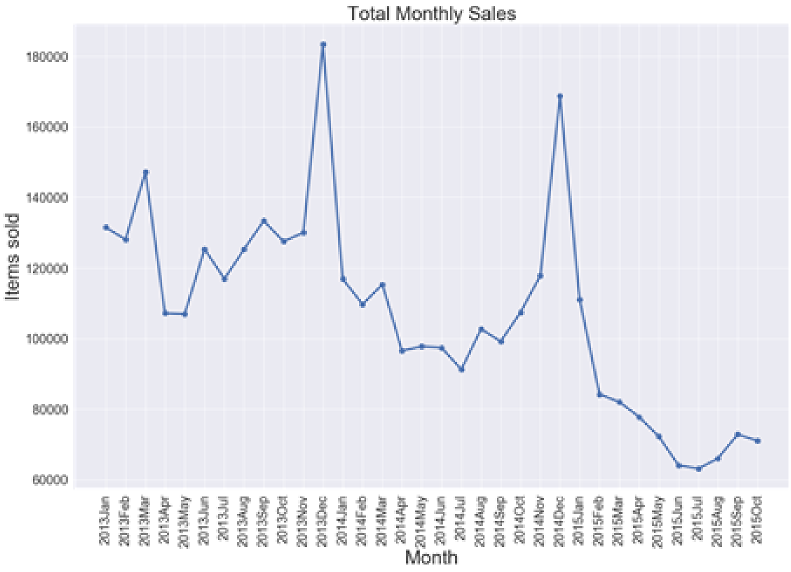
Description of Data

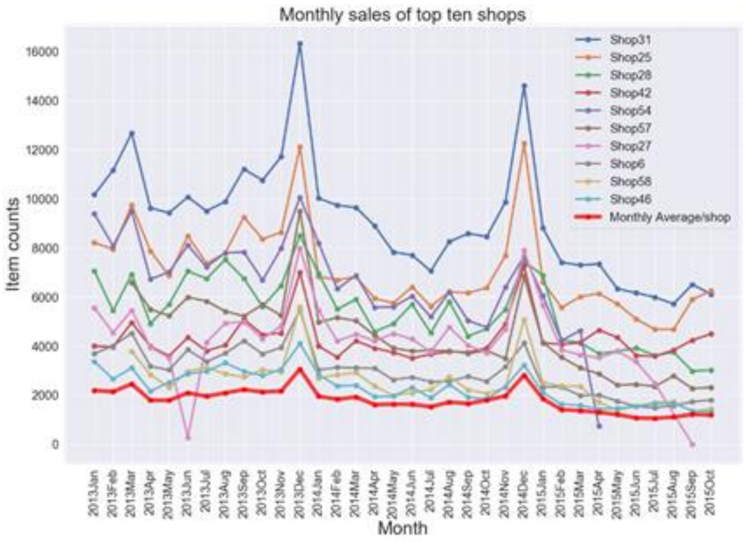
- Kaggle competition data set
- sales_train.csv with daily historical data for 33 months
- test.csv for test data to forecast the sales for next month
- items.csv with supplemental information about the items/products
- item_categories.csv with supplemental information about the items categories
- shops.csv with supplemental information about the shops

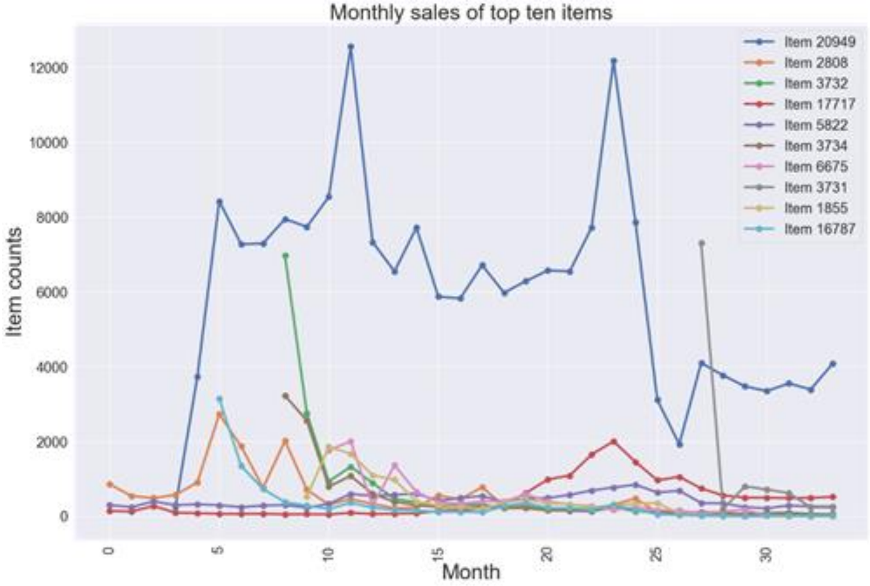
Data Wrangling

- Test file contains shop id and item id. Each shop and item combination is given a unique id and prediction is to be made about the sales of that unique id (meaning sales of that particular item from that particular shop)
- These information have been combined before proceeding.

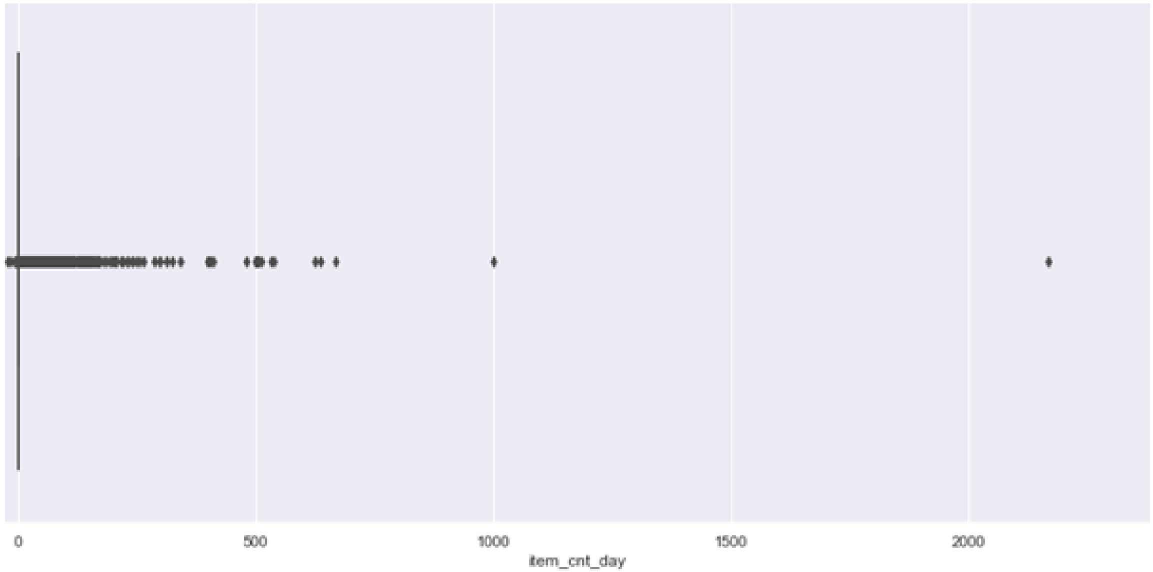
Exploratory Data Analysis

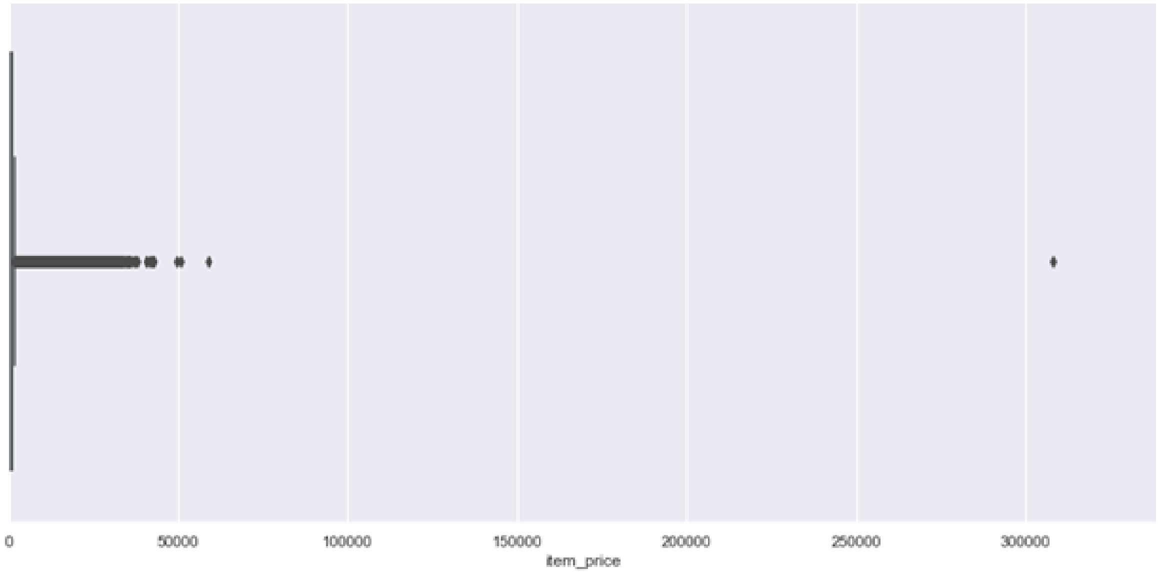






Exploratory Data Analysis (contd..)





Exploratory Data Analysis (contd..)

Feature Engineering

```python
# Add lag features, i.e. add data for one, two, three or more months earlier
# Here date_block_num comes handy
lag_feature_list = [1,2,3,6,12]

for lag in lag_feature_list:
    shifted_col_name = ('item_cnt_month_lag%s' % lag)
    train[shifted_col_name] = train.sort_values('date_block_num').groupby(['shop_id',
                                               'item_category_id', 'item_id'])['item_cnt_month'].shift(lag)
    train[shifted_col_name].fillna(0, inplace=True)

train.head(10).T
```

	0	1	2	3	4	5	6	7	8	9
date_block_num	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
shop_id	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
item_id	5572.0	5643.0	5583.0	7893.0	7894.0	7895.0	7956.0	1409.0	1467.0	3076.0
item_category_id	2.0	2.0	5.0	6.0	6.0	6.0	6.0	19.0	19.0	19.0
item_cnt_month	9.0	1.0	2.0	3.0	1.0	4.0	2.0	1.0	1.0	1.0
item_cnt_month_lag1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
item_cnt_month_lag2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
item_cnt_month_lag3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
item_cnt_month_lag6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
item_cnt_month_lag12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Data Modeling

- Linear Regression
- Ridge Regression
- Lasso Regression
- Decision Tree
- Bagging Regressor
- Random Forest
- Adaptive Boost Regressor
- Gradient Boost
- XGBoost
- LightGBM

Model Performance (with default parameters)

Algorithm	r^2	rmse
Gradient Boost	0.677656	1.505827
XGBoost	0.674566	1.500516
Linear Regression	0.661112	1.349141
LightGBM	0.653470	1.546350
Ridge Regression	0.629179	1.351676
Random Forest	0.626711	1.531725
Bagging Tree	0.625435	1.542834
Decision Tree	0.502483	1.934200
Lasso Regression	-0.000437	2.453177
Adaptive Boost	-3.260495	2.512668

Model Performace (with tuned parameters)

Algorithm	**Tuned r^2**	**Tuned rmse**
Gradient Boost	0.675227	1.461838
XGBoost	0.669124	1.468157
Random Forest	0.664986	1.457858
Bagging Tree	0.661905	1.425319
LightGBM	0.656737	1.533018

Final Prediction

- Picked Gradient Boost with default parameters as it had the best scores among all the models tested (with default or tuned parameters).

Future Directions

- As shown in EDA, the sales in general is in decline over time. It would be nice to have the data for later years to make a better understanding of there is an actual decline in sales of the items or something else.
- It would be nice to take into account some special events (like holidays, festivals etc).