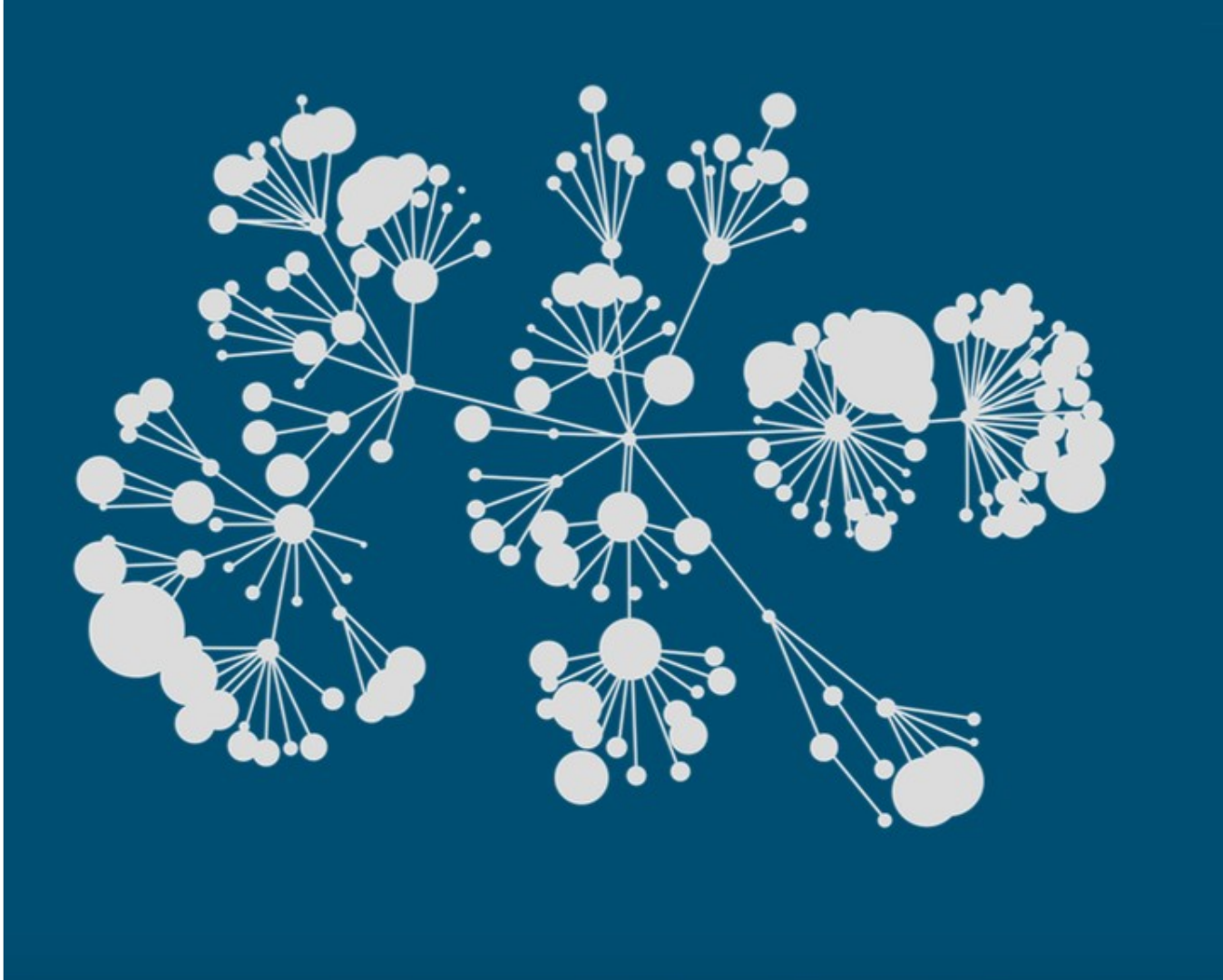
Kaggle

Shop sales prediction

kaggle



Agenda

1. Background
2. Summary
3. Feature selection & engineering
4. Training methods
5. Important findings
6. Simple model

Background

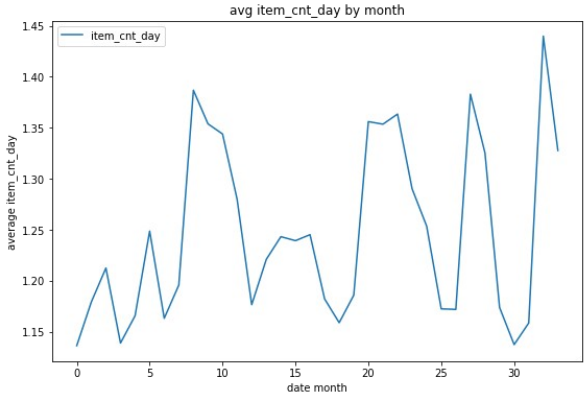
- Junior Researcher, Barcelona
- None, This is my first attempt. I learn machine learning in my spare time

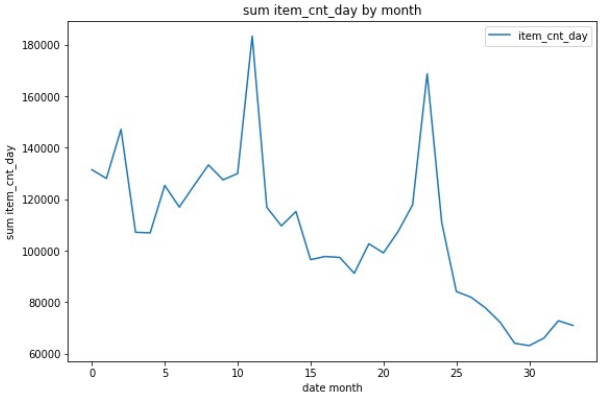
Summary

- Training methods:- XGBoost (successfully); LightGBM and Random forrest(unsuccessfully)
- The most important features were:-
  -item_category_id_avg_item_cnt_day_lag_1 (its a lag variable)
  - date_block_num
  -item_id_avg_item_price_lag_1 (another lag variable)
- Kaggle Python environment was used
- It takes around 2 hours with XGBoost to train the model

Features Selection / Engineering

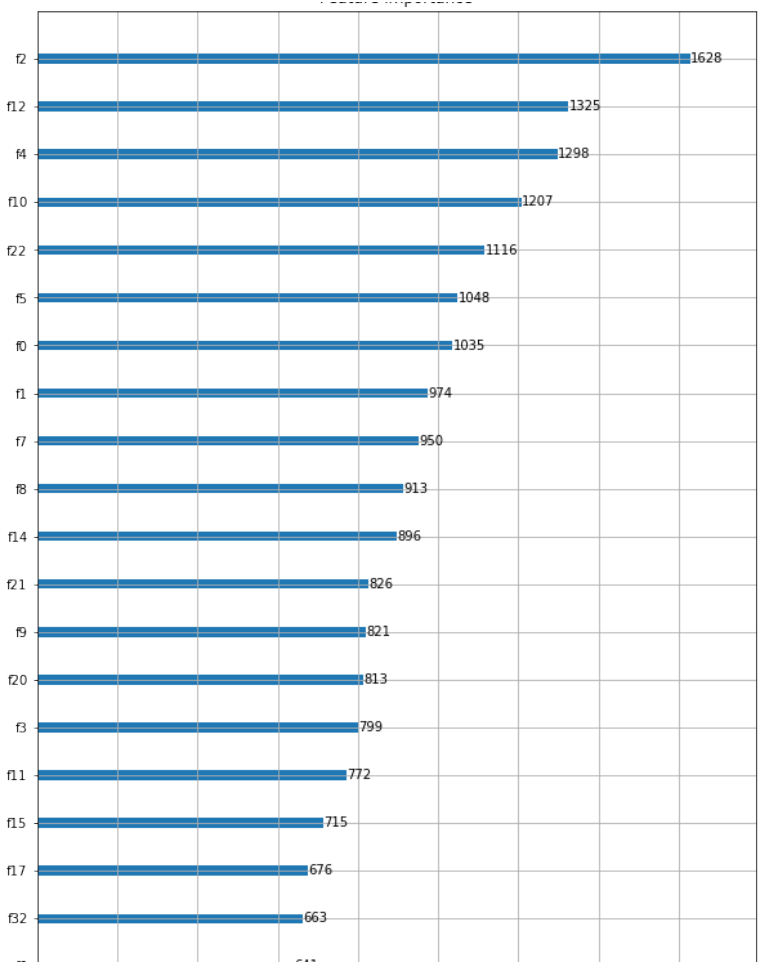
- Most important variables:-
  item_category_id_avg_item_cnt_day_lag_1 (its a lag variable)
  - date_block_num
  -item_id_avg_item_price_lag_1 (another lag variable)
- Data seasonality:-





Features Selection / Engineering

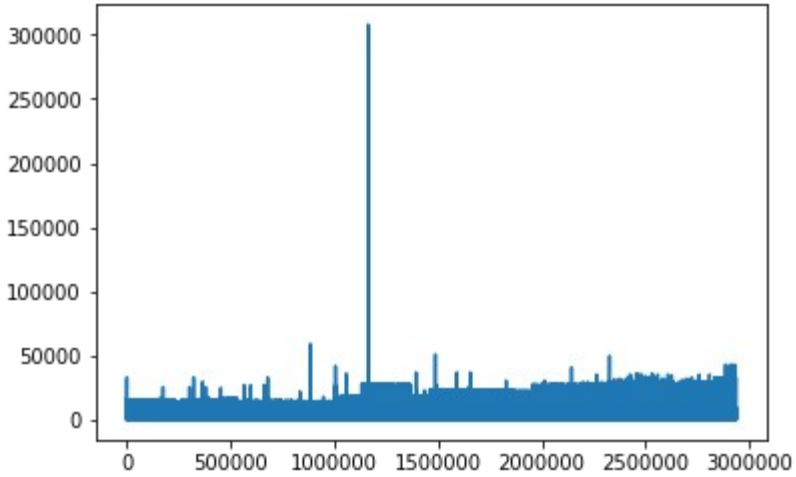
Variable Importance Plot



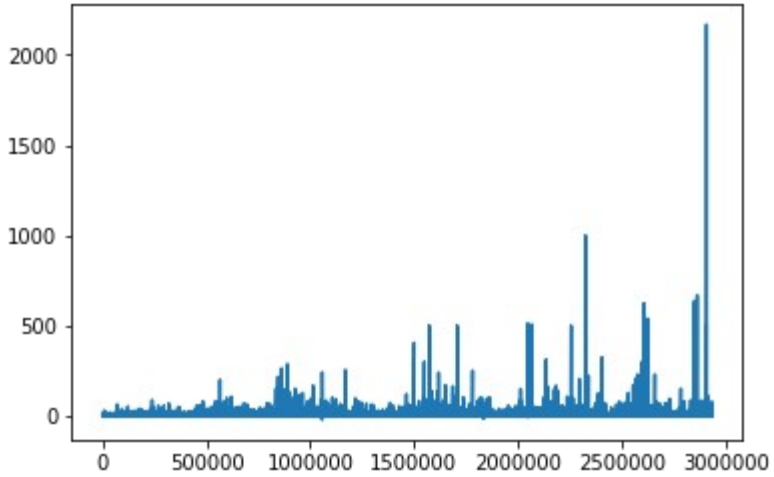
```
Index(['shop_id', 'item_id', 'item_category_id',
       'item_id_avg_item_price_lag_1', 'item_id_sum_item_cnt_day_lag_1',
       'item_id_avg_item_cnt_day_lag_1', 'shop_id_avg_item_price_lag_1',
       'shop_id_sum_item_cnt_day_lag_1', 'shop_id_avg_item_cnt_day_lag_1',
       'item_category_id_avg_item_price_lag_1',
       'item_category_id_sum_item_cnt_day_lag_1',
       'item_category_id_avg_item_cnt_day_lag_1', 'item_cnt_day_lag_1',
       'item_id_avg_item_price_lag_2', 'item_id_sum_item_cnt_day_lag_2',
       'item_id_avg_item_cnt_day_lag_2', 'shop_id_avg_item_price_lag_2',
       'shop_id_sum_item_cnt_day_lag_2', 'shop_id_avg_item_cnt_day_lag_2',
       'item_category_id_avg_item_price_lag_2',
       'item_category_id_sum_item_cnt_day_lag_2',
       'item_category_id_avg_item_cnt_day_lag_2', 'item_cnt_day_lag_2',
       'item_id_avg_item_price_lag_4', 'item_id_sum_item_cnt_day_lag_4',
       'item_id_avg_item_cnt_day_lag_4', 'shop_id_avg_item_price_lag_4',
       'shop_id_sum_item_cnt_day_lag_4', 'shop_id_avg_item_cnt_day_lag_4',
       'item_category_id_avg_item_price_lag_4',
       'item_category_id_sum_item_cnt_day_lag_4',
       'item_category_id_avg_item_cnt_day_lag_4', 'item_cnt_day_lag_4',
       'item_id_avg_item_price_lag_7', 'item_id_sum_item_cnt_day_lag_7',
       'item_id_avg_item_cnt_day_lag_7', 'shop_id_avg_item_price_lag_7',
       'shop_id_sum_item_cnt_day_lag_7', 'shop_id_avg_item_cnt_day_lag_7',
       'item_category_id_avg_item_price_lag_7',
       'item_category_id_sum_item_cnt_day_lag_7',
       'item_category_id_avg_item_cnt_day_lag_7', 'item_cnt_day_lag_7'],
      dtype='object')
```

Method

- Some data values which are outliers have been removed
- Only XGBoost was used with hyperparameters taken from hit and trial and other similar works (owing to less computational power)
- LGBM and RF were tried but did not yield good results. More resources are needed to completely exploit the search space for hyper-parameters for the two methods
- Mean encoding was used to generate variables using suggestions from the course itself



item_count_plot



item_price_plot

Outliers

Result

XGBoost was used until rmse has stopped decreasing or until a certain iterations

```
[0]     train-rmse:1.42962      valid-rmse:1.49058
Multiple eval metrics have been passed: 'valid-rmse' will be used for early
stopping.
Will train until valid-rmse hasn't improved in 50 rounds.
[50]    train-rmse:1.10402      valid-rmse:1.2261
[100]   train-rmse:1.05808      valid-rmse:1.18668
[150]   train-rmse:1.03716      valid-rmse:1.16694
[200]   train-rmse:1.02543      valid-rmse:1.16025
[250]   train-rmse:1.01713      valid-rmse:1.15682
[300]   train-rmse:1.00911      valid-rmse:1.15323
[350]   train-rmse:1.00293      valid-rmse:1.15006
[400]   train-rmse:0.9983       valid-rmse:1.14778
[450]   train-rmse:0.9937       valid-rmse:1.14646
[499]   train-rmse:0.989128     valid-rmse:1.1451
```

kaggle™