KAGGL E

predict Future sales

Team_KeSemi

정양섭 주혜린 송상민 백혜수 최준섭 박예원

2022. 07. 20

CONTENTS

PROJECT TRACKER 15 DAYS

NO. 01

PROJECT	TEAM	ke_Semi	
DESCRIPTION Kaggle_Predict Future Sales	DURATION	15 Days	DUE
	START	7/5	END 7/20

NOTES

Role

정양섭 : 팀장, 데이터전처리 부터 모델링 전반적인 부분 총괄

박예원 : 부팀장, 데이터 전처리, 이론 공부 및 모델링 파트 총괄

송상민 : 데이터 전처리

주혜린 : 데이터 전처리, ppt제작

최준섭 : 데이터 전처리, 모델링

백혜수 : 데이터 전처리, 모델링, ppt제작

TOP GOAL Predict Future Sales

SUB GOAL 1	SUB GOAL 2	SUB GOAL 3
Data Preprocessing	Modelling	Result Derivation

STEPS	DUE
☐ 문제이해(평가지표 파악)	7/5~6
☐ 데이터 분석 및 전처리	7/7~11
☐ 적합한 모델 찾기 적용 및 검증	7/12~17
☐ Hyperparameter Optimization	7/17~20
☐ 발표준비	7/18~20

ACTION PLAN	DAY 1	DAY 2	DAY 3	DAY 4	DAY 5	DAY 6	DAY 7	DAY 8	DAY 9	DAY 10	DAY 11	DAY 12	DAY 13	DAY 14	DAY 15
TASKS	5일	6일	7일	8일	9일	10일	11일	12일	13일	14일	15일	16일	17일	18일	19일
일정수립															
문제 이해															
전처리															
모델링															
성능개선															
발표자료제작															
코드 리뷰 및 스터디															

KeSemi

Summarize

01

▼

one of the largest Russian software firms - 1C Company

02

▼

Time Series Dataset

03

▼

predict total sales for every product and store in the next month

KeSemi

Main Data

STEP 01 STEP 02 STEP 03 STEP 04

Main Data Variables

KeSemi

Main Data

Data Name	Variables
item.csv	items
shops.csv	shops
sales_train.csv	train
item_categories.csv	item_categories
sample_submission.csv	submission

KeSemi

Preprocessed Data

STEP 01 STEP 02 STEP 03 STEP 04

preprocessed data

KeSemi

Preprocessed Data

Preprocessed Data	Mean
item_high_categories	아이템과 아이템 카테고리, 아이템 상위 카테고리를 가진 데이터 프레임
sales_df	최종적으로 전처리가 종료된 데이터

KeSemi

Added Functions

STEP 01 STEP 02 STEP 03 STEP 04

Added Functions

KeSemi

Added Functions

Added Functions	Mean
reduce_mem_usage	데이터 프레임의 용량을 줄려주는 함수
clean_text	특수문자를 제거하는 함수
price_item_cnt_day_boxplot	item_price, item_cnt_day의 박스 플롯을 만드는 함수
make_high_category	item_category를 분류하여 새로운 열을 만드는 함수
draw_x_group_y_sum_barplot	index는 x, values는 y의 합을 barplot으로 만드는 함수
add_mean_feature	base_feature_name를 기준으로 mean_feature_names의 값들을 한달 평균을 구해서 df에 병합해주는 함수
find_high_corr_location	상관계수가 높은 것들만 출력해주는 함수

KeSemi

Added Functions

```python
def add_mean_feature(df ,
                     base_feature_names ,
                     num_feature_name):
    '''
    base_feature_name를 기준으로 num_feature_name의 값의
    한달 평균을 구해서 df에 병합해주는 함수

    df 데이터가 있는 데이터 프레임
    base_feature_names 기준이 되는 특성이름
    num_feature_name 평균이 구해지는 열이름
    '''
    # 사용에 필요한 데이터만 복사
    df_temp = df[base_feature_names + [num_feature_name]].copy()

    # base_feature_names이 너무 많으면 오류를 일으킨다.
    if len(base_feature_names) not in range(1,4):
        raise
    # 새로 생성 되는 feature 의 이름 짓기
    if len(base_feature_names) == 1:
        feature_name = 'date_' +num_feature_name+'_mean'
    elif len(base_feature_names) == 2:
        feature_name = base_feature_names[1] +'_'+num_feature_name.split('_')[1]+ '_month_mean'
    else:
        feature_name = base_feature_names[1] + '_' + base_feature_names[2] +'_'+num_feature_name.split('_')[1]+'_month_mean'
    # num_feature_name >> feature_name 이름 바꾸기
    df_temp = df_temp.rename(columns = {num_feature_name:feature_name})
    print(f'{feature_name}이 생성되었습니다.')
    # base_feature_names을 기준으로 group 지어주고 feature_name의 평균을 구합니다.
    df_temp = df_temp.groupby(base_feature_names).agg({feature_name : np.mean}).reset_index()
    df = df.merge(df_temp ,
                  on =base_feature_names,
                  how= 'left')
    # df 와 feature_name을 반환하여 바로 add_lag_data을 돌릴수 있도록 합니다.
    return df , feature_name
```

KeSemi

Added Functions

```python
def add_lag_data(df ,lag_feature, lag_periods= [1] , drop = True):
    '''
    lag_period 만큼의 시차 데이터를 생성하는 함수

    df 데이터가 있는 데이터 프레임
    lag_feature 시차데이터가 생성될 열의 이름
    lag_periods= [1]  lag_periods 시차데이터를 얼마나 생성할지 정함
    drop = True  시차 데이터를 생성 후 원래의 데이터 제거 여부
    '''
    # 기준이 되는 특성이름 리스트
    base_feature_names = ['date_block_num','shop_id','item_id']
    # lag_periods값만 큼 시차 데이터를 생성한다.
    for i in lag_periods:
        # 필요한 데이터만 임시로 저장
        df_temp = df[base_feature_names + [lag_feature]].copy()
        # 생성되는 특성이름
        feature_name  = lag_feature +"_lag_"+str(i)
        # 생성되는 특성이름 적용
        df_temp.columns = base_feature_names +[feature_name]
        # date_block_num 값에 전체적으로 1값을 더하면서 shift를 한 것처럼 한다.
        df_temp['date_block_num'] +=i
        df = df.merge(df_temp,
                      on=base_feature_names,
                      how='left')
        # nan값을 0으로 채워준다.
        df[feature_name] = df[feature_name].fillna(0)
        print(f'{feature_name}을 생성하였습니다.')
        # 만약 drop =True 이면 lag_feature를 drop한다.
    if drop :
        df = df.drop(columns = [lag_feature] ,axis =1)

    return df
```

KeSemi

Added Variables

STEP 01 STEP 02 STEP 03 STEP 04

Added Variables

KeSemi

Added Variables

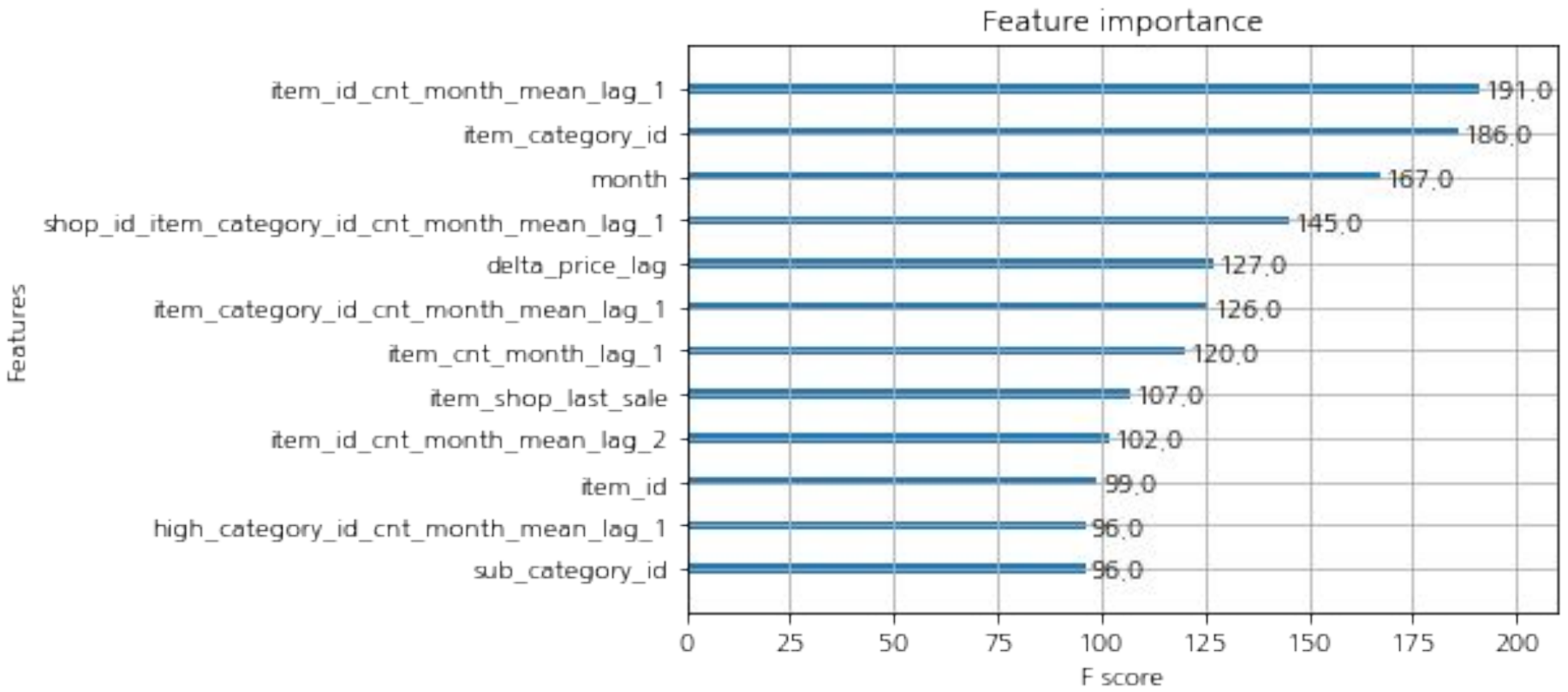
Added Variables	Mean
diff_train_test_shop_id	test 데이터의 shop_id가 train 데이터에 있는지 확인
diff_test_train_item_id	train 데이터의 item_id가 test 데이터에 있는지 확인
diff_test_items_item_id	items 데이터의 item_id가 test 데이터에 있는지 확인
temp_train	임시로 train값을 저장하기 위한 변수
train_item_cnt_month	한 달간 데이터를 종합할 때 사용한 변수
item_categories_value_counts	아이템 상위 카테고리의 value_counts를 저장하는 변수
base_feature_names	pivot 이나 Groupby를 할때 기준이 되는 열이름

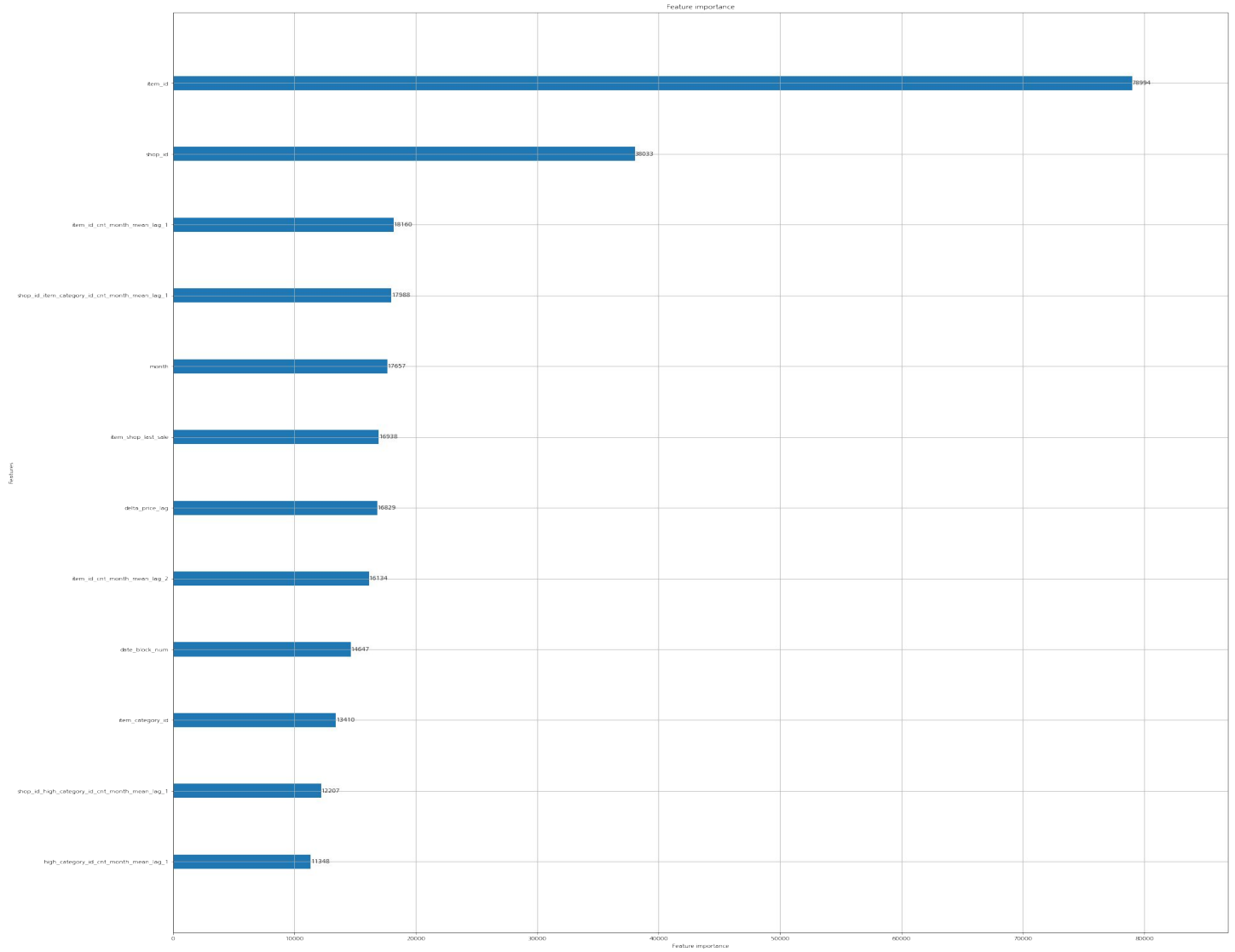
🚀XGBoost

- 💡 분류 및 회귀 문제에 모두 사용 가능
- 💡 Boosting 기법 + 병렬학습 지원
- 💡 과적합 규제

🥿LightGBM

- 🚿 빠른속도 & 메모리 사용량이 상대적으로 적은편
- 🚿 XGBoost의 단점 보완
- 🚿 대용량 데이터 처리에 효과적


Feature importance
item_id_cnt_month_mean_lag_1
item_category_id
month
shop_id_item_category_id_cnt_month_mean_lag_1
delta_price_lag
item_category_id_cnt_month_mean_lag_1
item_cnt_month_lag_1
item_shop_last_sale
item_id_cnt_month_mean_lag_2
item_id
high_category_id_cnt_month_mean_lag_1
sub_category_id
191.0
186.0
167.0
145.0
127.0
126.0
120.0
107.0
102.0
99.0
96.0
96.0
Features
0
25
50
75
100
125
150
175
200
F score


Feature importance
item_id
shop_id
item_id_cnt_month_mean_lag_1
shop_id_item_category_id_cnt_month_mean_lag_1
month
item_shop_last_sale
delta_price_lag
item_id_cnt_month_mean_lag_2
date_block_num
item_category_id
shop_id_high_category_id_cnt_month_mean_lag_1
high_category_id_cnt_month_mean_lag_1
78994
36033
18160
17988
17657
16938
16829
16134
14647
13410
12207
11348
Features
0
10000
20000
30000
40000
50000
60000
70000
80000
Feature importance

👀 Learned from the Project

Reference

시계열 데이터 정의와 시계열 자료 분석 방법 – 전통적방법
[Predict Future Sales] playground 커널 리뷰 1
python random모듈 3개 정리 (randint, rand, randn)
민감도(Sensitivity)와 특이도(Specificity)
Ridge regression(릿지 회귀)와 Lasso regression(라쏘 회귀) 쉽게 이해하기
Introduction to ARIMA: nonseasonal models
Feature engineering, xgboost | Kaggle

KeSemi

Code Review