개인 과제 레포트

제출 일자 : 2021.02.16.(화)

경상대학교 수학과 허지혜

1. 주제

Jane Street Market Prediction

2. 목표

Jane Street에서 제공하는 여러 익명화된 기능 집합을 가지고 거래 기회를 예측한다.

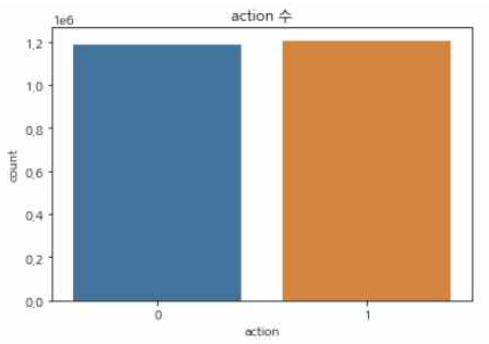
3. 데이터

example_sample_submission.csv : 제출 파일
example_test.csv : 모의 data set
features.csv : 익명화된 기능과 관련된 메타 데이터
train.csv : the training set, 기록 데이터 및 반환 값 포함 (2390491,138)

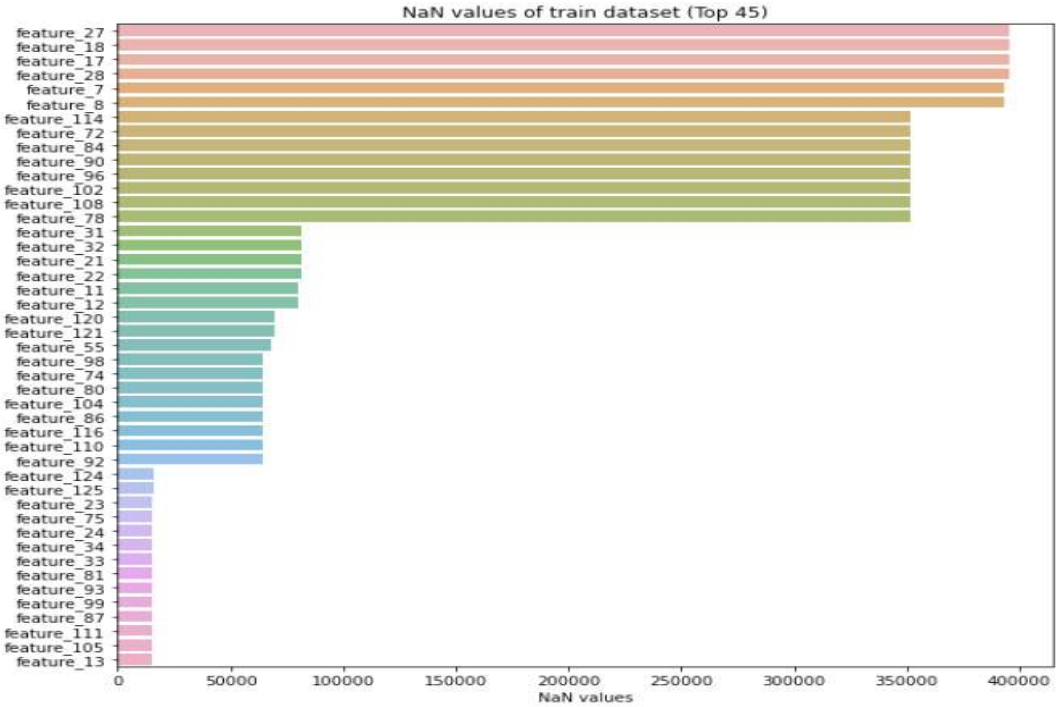
- date
- weight
- resp_{1,2,3,4}
- feature_{0,1,...,129}
- ts_id

4. EDA

실제 주식 시장 데이터를 나타내는 익명화 된 기능 집합 feature_{0,1,...129}
각 행은 거래 기회를 나타내며 예측할 값은 거래 유무이다.
1은 거래함, 0은 거래 안함을 나타낸다.
각 거래에는 관련 가중치와 응답이 있으며 거래 수익을 나타낸다.
날짜 열은 거래일을 나타내는 정수이고 ts_id는 시간을 나타낸다.
train.csv에는 action에 대한 열이 없으니 만들어주자.



train['resp']를 기준으로
resp 양수 : action 1, resp 음수 : action 0
으로 나타낸 후 값을 살펴보면 위와 같다.
예측할 값이 0과 1인 이중 분류이다.

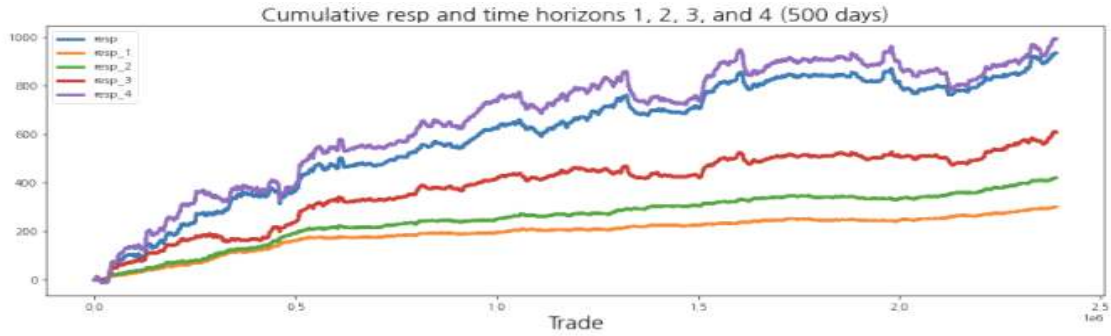


결측값이 있는 열들이 많아서 결측값은 평균으로 처리를 해주었다.

```
# 잘 처리되었다.
df_train.isnull().sum()
```

```
date           0
weight         0
resp_1         0
resp_2         0
resp_3         0
              ..
feature_127    0
feature_128    0
feature_129    0
ts_id          0
action         0
Length: 139, dtype: int64
```

또한 가중치가 0인 거래는 데이터에 기여 하지 않기 때문에 삭제해주었다.
제거 결과 1981287개의 행과 139개의 열이 남았다.



resp_{1,2,3,4}와 resp 에 대하여 각각 누적합을 구해보았다.
resp(파랑)가 resp_4(보라)와 가장 가깝게 따르는 모습을 확인할 수 있다.

5. Modeling

분류 모델이므로 xgboost를 썼다.
X = feature_{0,...129}
y = action

```
from sklearn.model_selection import train_test_split
X = df_train.loc[:,df_train.columns.str.contains('feature')]
y = df_train.loc[:,'action']

X_train, X_valid, y_train, y_valid = train_test_split(X,y,test_size=0.2, random_stat
```

```
import xgboost as xgb
dtrain = xgb.DMatrix(X_train, label=y_train)
dvalid = xgb.DMatrix(X_valid, label=y_valid)
```

```
clf = xgb.XGBClassifier()
```

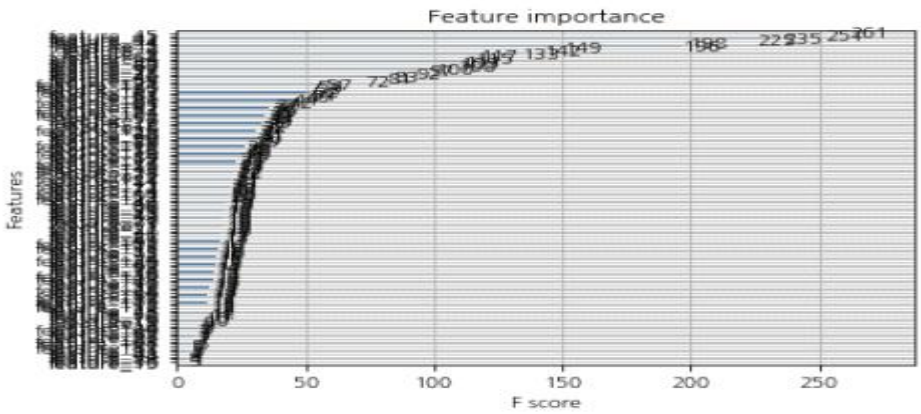
```
clf.fit(X_train, y_train)
```

```
C:\Users\hu612\Anaconda3\lib\site-packages\xgboost\sklearn.py:888: UserWarning: The
use of label encoder in XGBClassifier is deprecated and will be removed in a future
release. To remove this warning, do the following: 1) Pass option use_label_encoder
=False when constructing XGBClassifier object; and 2) Encode your labels (y) as int
egers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)

[16:19:05] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.3.0/sr
c/learner.cc:1061: Starting in XGBoost 1.3.0, the default evaluation metric used wi
th the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitl
y set eval_metric if you'd like to restore the old behavior.

XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
              learning_rate=0.300000012, max_delta_step=0, max_depth=6,
              min_child_weight=1, missing=nan, monotone_constraints='()',
              n_estimators=100, n_jobs=8, num_parallel_tree=1, random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
              tree_method='exact', validate_parameters=1, verbosity=None)
```

feature importance



Mean Absolute Error : 0.449161

해야할 것

1. 분류 모델 더 써보기
2. gridsearchCV말고도 찾는 방법 많던데 하나 써보기
3. 시계열데이터 모델에 적용시켜보기
4. feature importance로 나온거 상위 몇 개로만 돌려보거나, pca 써서 x 개수 줄여서 돌려보기

참고

https://www.kaggle.com/carlmcbrideellis/jane-street-eda-of-day-0-and-feature-importance

https://www.kaggle.com/christopherworley/jane-street-lstm