주식 예측 프로젝트

- 프로그래밍 언어 및 도구: Python (pandas, numpy, matplotlib, imblearn, sickit-learn, xgboost)
- 활용한 기술: EDA, 데이터 전처리, 머신러닝

[프로젝트 소개]

- 2024년도 봄 학기 머신러닝 수업 프로젝트
- 나스닥에 상장된 기업들의 1년치 재무 지표를 활용하여, 지난 1년간 기업의 주가가 상승했는지 하락했는지를 예측했습니다.

Step 1: EDA

[데이터 확인]

	Name	Revenue	Revenue Growth	Cost of Revenue	Gross Profit	R&D Expenses	SG&A Expense	operatingProfitMargin
0	SBFG	4.520300e+07	0.0514	0.000000e+00	4.520300e+07	0.000000e+00	2.754700e+07	1.0
1	FOMX	3.669000e+06	-0.3362	1.300000e+04	3.656000e+06	5.777900e+07	1.149100e+07	1.0
2	VIA	1.326300e+10	NaN	7.436000e+09	5.827000e+09	NaN	3.005000e+09	NaN
3	ABM	5.453600e+09	0.0600	4.881200e+09	5.724000e+08	0.000000e+00	4.366000e+08	1.0
4	THS	6.307100e+09	0.0214	5.226700e+09	1.080400e+09	0.000000e+00	7.007000e+08	1.0
5	UUUU	3.104600e+07	-0.4309	2.271000e+07	8.336000e+06	0.000000e+00	1.492300e+07	1.0
6	JMU	8.873655e+07	0.2122	8.818778e+07	5.487680e+05	0.000000e+00	2.190326e+07	1.0
7	CBH	NaN	NaN	NaN	NaN	NaN	NaN	NaN

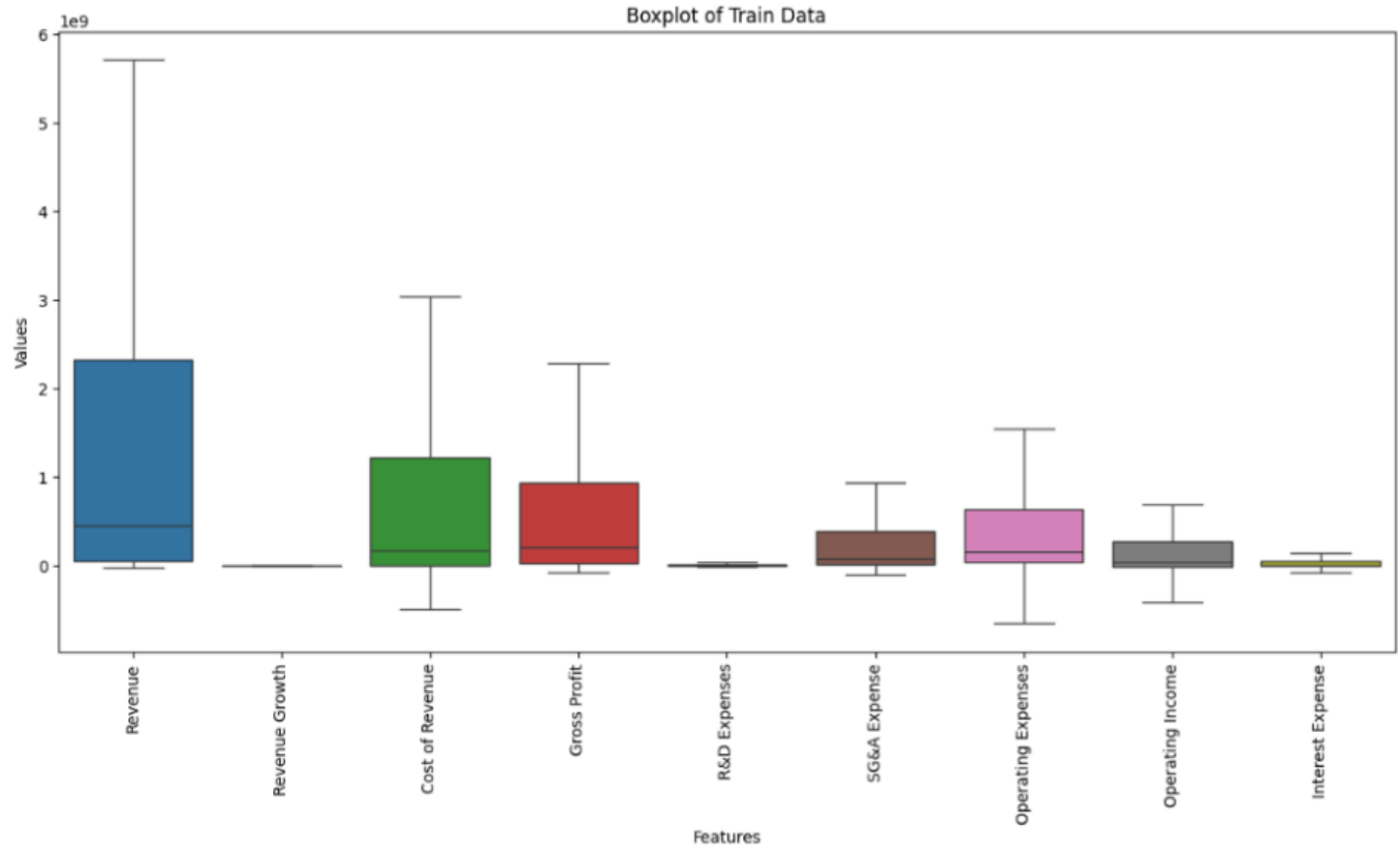
...

Debt Growth	R&D Expense Growth	SG&A Expenses Growth	Sector	Class
-0.0729	0.0000	0.0096	Financial Services	0
-1.0000	1.2311	0.2462	Healthcare	0
NaN	NaN	NaN	Consumer Cyclical	0
3.3914	0.0000	0.0646	Industrials	0
-0.0879	0.0000	-0.0592	Consumer Defensive	1
-0.0698	0.0000	-0.0384	Basic Materials	1
0.0000	0.0000	-0.2160	Consumer Cyclical	0
NaN	NaN	NaN	Financial Services	0

- 3,138개의 기업과 229개의 변수가 있습니다.
- 'Name' 변수는 머신러닝 모델에서 중요한 역할을 하지 않습니다.
- 'Sector' 변수는 유일한 범주형 변수이며, 나머지 변수들은 모두 숫자형 변수입니다.
- 'Class' 변수는 목적 변수입니다.
- 각 변수마다 NA 값들이 존재합니다.
    - 몇몇 회사들(CBH 등)은 'Name', 'Sector', 'Class'를 제외하고는 아무 값도 입력되지 않아 대처가 필요합니다.
- 재무 지표에서 'operatingProfitMargin'은 % 형식으로 표기되어야 하지만, 데이터셋에는 모두 1 또는 NA 값으로만 표기되어 있어 제거가 필요합니다.

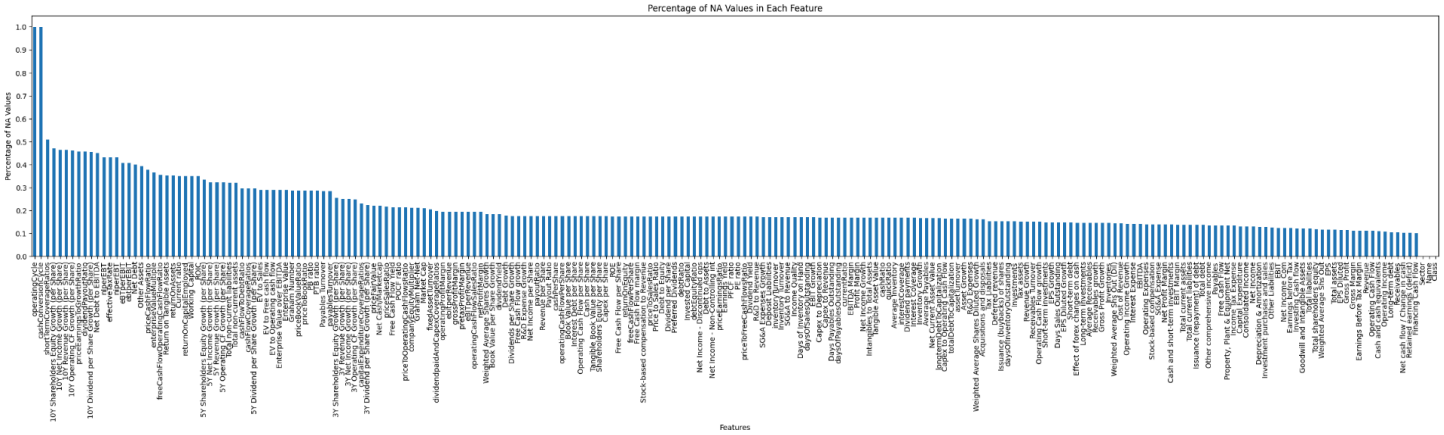
[통계 요약]

	Revenue	Revenue Growth	Cost of Revenue	Gross Profit	R&D Expenses	SG&A Expense	Operating Expenses	Operating Income	Interest Expense	Earnings before Tax
count	3.075000e+03	2939.000000	2.964000e+03	3.068000e+03	2.897000e+03	2.983000e+03	2.980000e+03	3.088000e+03	2.974000e+03	3.038000e+03
mean	5.361782e+09	1.169073	3.389896e+09	2.059012e+09	1.161536e+08	8.710404e+08	1.411698e+09	6.233607e+08	1.082823e+08	5.054936e+08
std	3.531643e+10	22.485009	2.959656e+10	8.479467e+09	8.909429e+08	3.198456e+09	5.385638e+09	3.044427e+09	4.742665e+08	2.451799e+09
min	-3.004500e+07	-2.125700	-2.986888e+09	-7.321000e+07	-7.716687e+06	-1.043667e+08	-1.088000e+09	-1.868300e+10	-1.176690e+08	-1.819800e+10
25%	5.900000e+07	0.000000	2.948825e+06	3.363150e+07	0.000000e+00	1.902400e+07	3.780130e+07	-4.413353e+06	0.000000e+00	-9.542500e+06
50%	4.614870e+08	0.072500	1.741980e+08	2.048185e+08	0.000000e+00	8.407700e+07	1.638455e+08	3.984300e+07	5.108000e+06	2.640257e+07
75%	2.326197e+09	0.205400	1.220002e+09	9.438390e+08	1.432700e+07	3.862010e+08	6.440122e+08	2.715873e+08	5.728800e+07	2.030585e+08
max	1.695864e+12	825.959800	1.465577e+12	2.302870e+11	2.262000e+10	5.343600e+10	8.117715e+10	6.513565e+10	1.340100e+10	6.408900e+10



- 통계 요약 표와 박스 플롯을 보면 변수마다 값의 기준이 매우 크게 차이난다는 것을 알 수 있습니다.
- 변수마다 값의 기준이 차이가 많이 나면 머신러닝 모델의 안정성에 영향을 줄 수 있기 때문에 대처가 필요합니다.

[결측치]

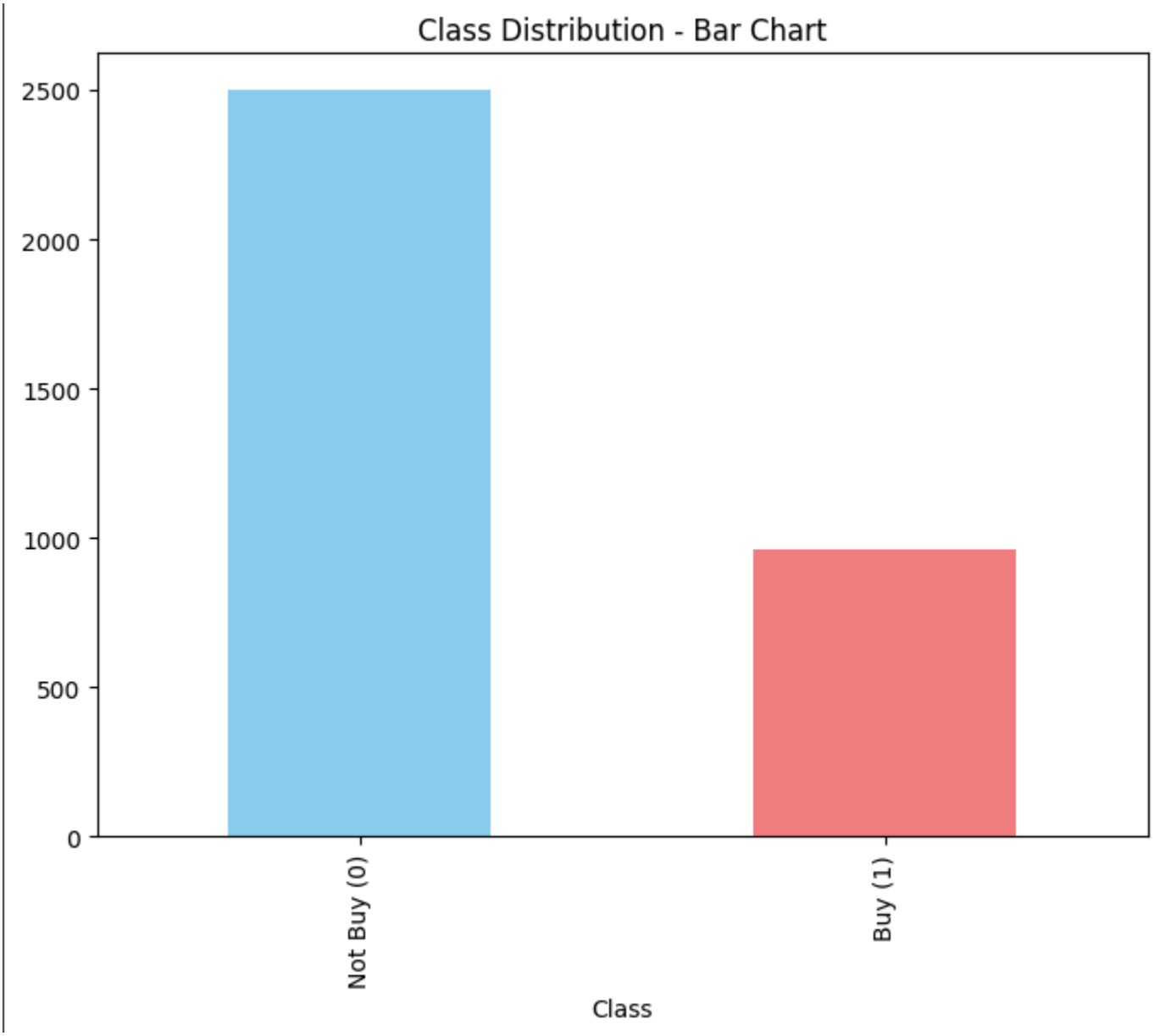


operatingCycle	0.999133
cashConversionCycle	0.999133
shortTermCoverageRatios	0.507950
10Y Shareholders Equity Growth (per Share)	0.469500
10Y Net Income Growth (per Share)	0.461983
10Y Revenue Growth (per Share)	0.461983
10Y Operating CF Growth (per Share)	0.460249
priceEarningsToGrowthRatio	0.455045
dividendPayoutRatio	0.455045
10Y Dividend per Share Growth (per Share)	0.453599
Net Debt to EBITDA	0.449552
nIperEBT	0.432206
effectiveTaxRate	0.432206
niperEBT	0.432206
eBTperEBIT	0.405898
ebtperEBIT	0.405898
Net Debt	0.398959
Other Assets	0.392888
priceCashFlowRatio	0.376698
enterpriseValueMultiple	0.364267
Sector	0.000000
Name	0.000000
Class	0.000000

⋮

- 'Sector', 'Name', 'Class'를 제외하고 모든 변수들이 결측치를 가지고 있습니다.
- 결측치는 머신러닝 성능에 영향을 주기 때문에 이에 대한 대처가 필요합니다.

[목적변수의 불균형]



- 목적 변수의 불균형은 머신러닝 모델이 소수의 목적 변수 값에 과적합되어 성능을 저하시킬 수 있습니다.
- 이번 프로젝트에서는 목적 변수의 분포에 상대적으로 덜 민감한 AUC-ROC를 사용하기 때문에 따로 전처리는 하지 않습니다.

Step 2: 데이터 전처리

[결측치 제거 및 처리]

7	CBH	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Financial Services	0

```
Number of columns without any info (all NA): 221
```

- 'Name', 'Sector', 'Class'를 제외한 모든 변수에 NA 값이 있는 221개의 기업들을 제거했습니다.

```
operatingCycle                0.999133
cashConversionCycle           0.999133
shortTermCoverageRatios       0.507950
```

operatingProfitMargin
1.0
1.0
NaN
1.0
1.0
1.0
1.0
NaN

- 변수의 결측치가 50% 이상인 'operatingCycle', 'cashConversionCycle', 'shortTermCoverageRatios'와 변수의 값이 잘못 입력된 것으로 추정되는 'operatingProfitMargin'을 제거했습니다.

Sector
Financial Services
Healthcare
Consumer Cyclical
Industrials
Consumer Defensive
...
Healthcare
Financial Services
Energy
Industrials
Energy

→

Sector_Basic Materials	Sector_Communication Services	Sector_Consumer Cyclical	Sector_Consumer Defensive	Sector_Energy
False	False	False	False	False
False	False	False	False	False
False	False	True	False	False
False	False	False	False	False
False	False	False	True	False
...	...	...	...	...
False	False	False	False	False
False	False	False	False	False
False	False	False	False	True
False	False	False	False	False
False	False	False	False	True

- KNN 알고리즘을 사용하여 결측치를 처리하기 위해 범주형 변수인 'Sector'를 이진 벡터 형태로 변환했습니다.
- 또한 머신러닝을 사용하기 위해 모든 범주형 변수를 이진 벡터 형태로 변환해야 합니다.

Revenue	0.111015
Revenue Growth	0.150332
Cost of Revenue	0.143105
Gross Profit	0.113038
R&D Expenses	0.162475
SG&A Expense	0.137612
Operating Expenses	0.138479
Operating Income	0.107256
Interest Expense	0.140214
Earnings before Tax	0.121711
Income Tax Expense	0.133565
Net Income - Non-Controlling int	0.172015
Net Income - Discontinued ops	0.172015
Net Income	0.128939
Preferred Dividends	0.172015
Net Income Com	0.122001
EPS	0.115062
EPS Diluted	0.114195
Weighted Average Shs Out	0.115640
Weighted Average Shs Out (Dil)	0.143394
Sector	0.000000
Class	0.000000

→

Revenue	0.0
Revenue Growth	0.0
Cost of Revenue	0.0
Gross Profit	0.0
R&D Expenses	0.0
SG&A Expense	0.0
Operating Expenses	0.0
Operating Income	0.0
Interest Expense	0.0
Earnings before Tax	0.0
Income Tax Expense	0.0
Net Income - Non-Controlling int	0.0
Net Income - Discontinued ops	0.0
Net Income	0.0
Preferred Dividends	0.0
Net Income Com	0.0
EPS	0.0
EPS Diluted	0.0
Weighted Average Shs Out	0.0
Weighted Average Shs Out (Dil)	0.0
Class	0.0
Sector_Basic Materials	0.0
Sector_Communication Services	0.0
Sector_Consumer Cyclical	0.0
Sector_Consumer Defensive	0.0
Sector_Energy	0.0
Sector_Financial Services	0.0
Sector_Healthcare	0.0
Sector_Industrials	0.0
Sector_Real Estate	0.0
Sector_Technology	0.0
Sector_Utilities	0.0

- KNNImputer는 결측값의 5개의 최근접 이웃을 찾아 이웃들의 평균값으로 결측값을 대체합니다. 이를 통해 데이터의 패턴을 유지하며 결측값을 처리할 수 있습니다.

[데이터 표준화]

	Revenue	Revenue Growth	Cost of Revenue	Gross Profit	R&D Expenses	SG&A Expense	Operating Expenses	Operating Income	Interest Expense	Earnings before Tax
count	3.075000e+03	2939.000000	2.964000e+03	3.068000e+03	2.897000e+03	2.983000e+03	2.980000e+03	3.088000e+03	2.974000e+03	3.038000e+03
mean	5.361782e+09	1.169073	3.389896e+09	2.059012e+09	1.161536e+08	8.710404e+08	1.411698e+09	6.233607e+08	1.082823e+08	5.054936e+08
std	3.531643e+10	22.485009	2.959656e+10	8.479467e+09	8.909429e+08	3.198456e+09	5.385638e+09	3.044427e+09	4.742665e+08	2.451799e+09
min	-3.004500e+07	-2.125700	-2.986888e+09	-7.321000e+07	-7.716687e+06	-1.043667e+08	-1.088000e+09	-1.868300e+10	-1.176690e+08	-1.819800e+10
25%	5.900000e+07	0.000000	2.948825e+06	3.363150e+07	0.000000e+00	1.902400e+07	3.780130e+07	-4.413353e+06	0.000000e+00	-9.542500e+06
50%	4.614870e+08	0.072500	1.741980e+08	2.048185e+08	0.000000e+00	8.407700e+07	1.638455e+08	3.984300e+07	5.108000e+06	2.640257e+07
75%	2.326197e+09	0.205400	1.220002e+09	9.438390e+08	1.432700e+07	3.862010e+08	6.440122e+08	2.715873e+08	5.728800e+07	2.030585e+08
max	1.695864e+12	825.959800	1.465577e+12	2.302870e+11	2.262000e+10	5.343600e+10	8.117715e+10	6.513565e+10	1.340100e+10	6.408900e+10

	Revenue	Revenue Growth	Cost of Revenue	Gross Profit	R&D Expenses	SG&A Expense	Operating Expenses	Operating Income	Interest Expense	Earnings before Tax
count	3.138000e+03	3.138000e+03	3.138000e+03	3.138000e+03	3.138000e+03	3.138000e+03	3.138000e+03	3.138000e+03	3.138000e+03	3.138000e+03
mean	-2.490749e-17	-1.132159e-17	5.660793e-18	-3.000220e-17	-1.132159e-18	9.623348e-18	-2.320925e-17	2.151101e-17	-2.745485e-17	1.641630e-17
std	1.000159e+00	1.000159e+00	1.000159e+00	1.000159e+00	1.000159e+00	1.000159e+00	1.000159e+00	1.000159e+00	1.000159e+00	1.000159e+00
min	-1.516369e-01	-1.531885e-01	-2.183848e-01	-2.503934e-01	-1.464555e-01	-3.051671e-01	-4.692980e-01	-6.388117e+00	-4.803396e-01	-7.739412e+00
25%	-1.491684e-01	-5.648422e-02	-1.145338e-01	-2.377384e-01	-1.375605e-01	-2.665151e-01	-2.574175e-01	-2.053768e-01	-2.278027e-01	-2.084644e-01
50%	-1.381855e-01	-5.298126e-02	-1.087804e-01	-2.178193e-01	-1.375605e-01	-2.462633e-01	-2.340794e-01	-1.906236e-01	-2.168079e-01	-1.934509e-01
75%	-8.586610e-02	-4.631883e-02	-7.341695e-02	-1.335974e-01	-1.187673e-01	-1.543723e-01	-1.437477e-01	-1.165563e-01	-1.091032e-01	-1.232415e-01
max	4.835525e+01	3.751883e+01	5.077360e+01	2.719417e+01	2.593644e+01	1.668183e+01	1.502771e+01	2.135580e+01	2.853294e+01	2.632945e+01

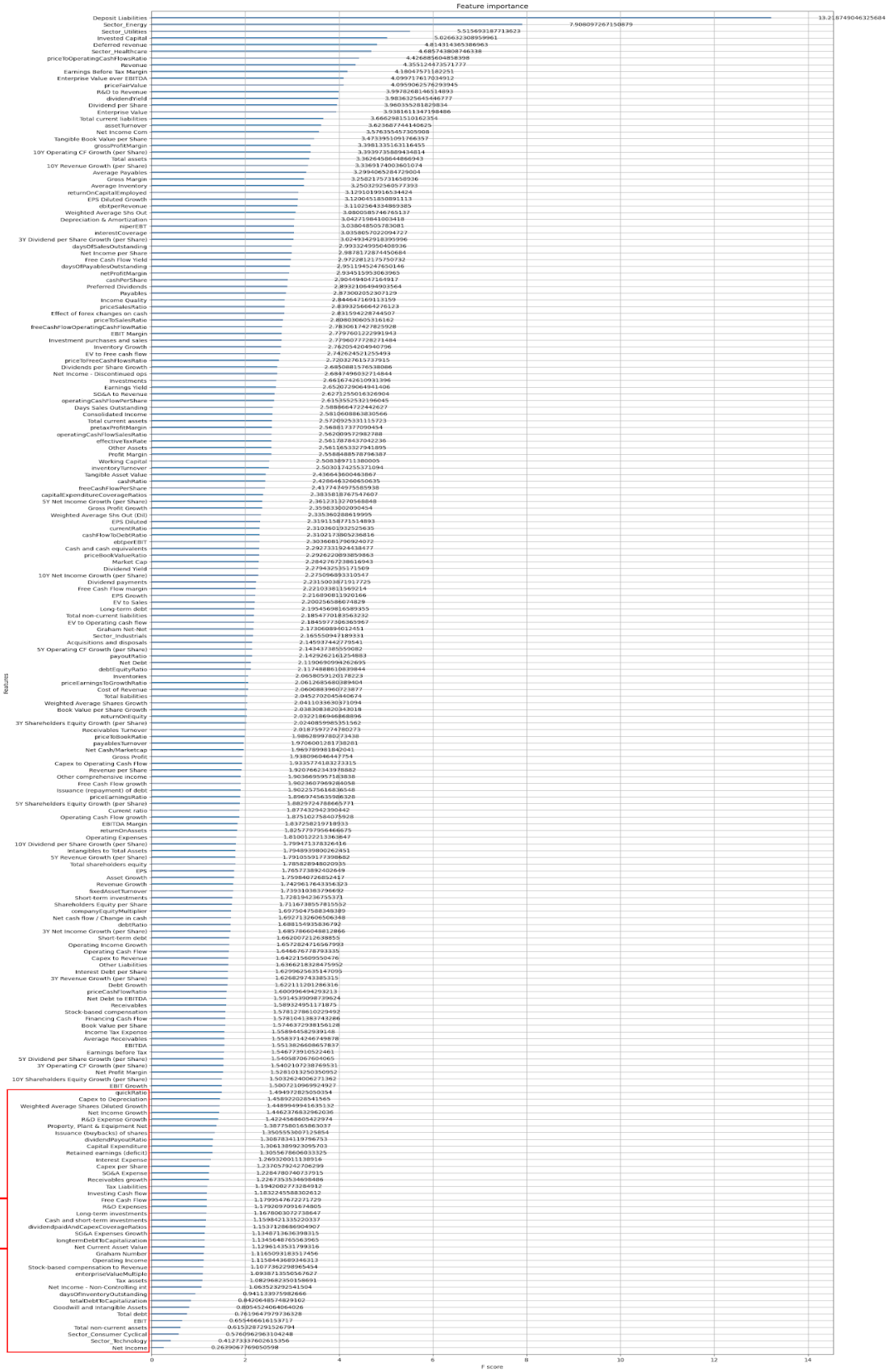
- 추후에 결정 트리 기반 모델을 사용할 예정이기 때문에 표준화가 필수는 아니지만, 변수가 많고 변수마다 값의 기준이 크게 차이나기 때문에 표준화를 통해 모델의 안정성을 향상시켰습니다.

[데이터 분할]

```
X_train = fr_final_train.drop('Class', axis=1)
y_train = fr_final_train['Class']
```

- 머신러닝 학습을 위해 학습 데이터셋을 설명변수만 있는 X_train 데이터셋과 목적변수만 있는 y_train 데이터셋으로 분할했습니다.

[변수 선택]



- 하이퍼파라미터 튜닝이 되지 않은 XGBoost 모델에 전처리가 완료된 데이터셋을 학습시킨 후, Gini feature importance 플롯을 생성하여 Gini 점수가 1.5 이하인 특징들을 제거하여 모델 성능을 최적화했습니다.

Step 3: 머신러닝 **(XGBoost)**

[하이퍼파라미터 튜닝]

```
from sklearn.model_selection import GridSearchCV

param_grid = {
    'learning_rate': [0.01, 0.05, 0.1],
    'n_estimators': [100, 300, 500],
    'max_depth': [2, 4, 6]
}

grid_search = GridSearchCV(xgb, param_grid, cv=3, n_jobs=-1, scoring='roc_auc', verbose = 1)

grid_search.fit(X_train_fr, y_train)

best_parameters = grid_search.best_params_
```

- XGBoost의 파라미터들을 여러 값으로 설정:
  - learning_rate: 모델 가중치를 업데이트하는 스텝 크기; 큰 값은 빠르게 학습하지만 덜 세밀함.
  - max_depth: 모델의 복잡성을 증가시켜 더 세밀한 패턴을 학습 가능.
  - n_estimators: 앙상블의 트리 수; 각 트리는 이전 트리의 오류를 수정.
- GridSearchCV를 통해 모든 조합을 테스트하여 가장 높은 roc_auc 점수가 나온 파라미터 조합을 저장합니다.

[예측]

```
best_model = grid_search.best_estimator_

y_pred = best_model.predict(fr_fr_final_test)
```

- roc_auc 점수가 가장 높게 나온 파라미터 조합을 사용하여 테스트 데이터셋의 결과 값을 예측합니다.

[점수 **(ROC AUC)**]

- ROC AUC의 값이 0.61로, 이는 모델이 실제로 주가가 오른 기업을 61%의 확률로 올바르게 예측하고, 39%의 확률로 잘못 예측한다는 것을 의미합니다.
- 모델이 무작위 예측보다는 약간 더 나은 성능을 보이지만, 실제 거래에서 사용할 만큼 좋은 성능은 아닙니다.
- 투자자들의 심리, 시장의 변동성, 예기치 않은 사건 등이 주가에 큰 영향을 미치기 때문에 단순한 수학적 모델링 만으로는 주가의 변동을 예측하기 어렵습니다.