KAIST MFE, 2025 Spring

Kim Hyeonghwan

2025-03-05

Table of contents

Welcome!

안녕하세요, KAIST MFE 25년 봄학기에 이수한 과목의 과제 등을 정리해두었습니다.

Part I

머신러닝2('25 봄)

1 빅데이터와 금융자료 분석 CH1

```
import numpy as np
import pandas as pd
```

```
missdict = {'f1': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
            'f2': [10., None, 20., 30., None, 50., 60., 70., 80., 90.],
            'f3': ['A', 'A', 'A', 'A', 'B', 'B', 'B', 'B', 'C', 'C']}
missdata = pd.DataFrame( missdict )
missdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   f1      10 non-null     int64
 1   f2      8 non-null      float64
 2   f3      10 non-null     object
dtypes: float64(1), int64(1), object(1)
memory usage: 368.0+ bytes
```

```
missdata.isna().mean()
```

```
f1    0.0
f2    0.2
f3    0.0
dtype: float64
```

```
tmpdata1 = missdata.dropna()
tmpdata1
```

	f1	f2	f3
0	1	10.0	A
2	3	20.0	A
3	4	30.0	A
5	6	50.0	B
6	7	60.0	B
7	8	70.0	B
8	9	80.0	C
9	10	90.0	C

```
tmpdata2 = missdata.dropna( subset=['f3'] )
tmpdata2
```

	f1	f2	f3
0	1	10.0	A
1	2	NaN	A
2	3	20.0	A
3	4	30.0	A
4	5	NaN	B
5	6	50.0	B
6	7	60.0	B
7	8	70.0	B
8	9	80.0	C
9	10	90.0	C

```
numdata = missdata.select_dtypes(include=['int64', 'float64'])
tmpdata3 = numdata.fillna( -999, inplace=False )
tmpdata3.describe()
```

	f1	f2
count	10.00000	10.000000
mean	5.50000	-158.800000
std	3.02765	443.562297
min	1.00000	-999.000000
25%	3.25000	12.500000
50%	5.50000	40.000000
75%	7.75000	67.500000

	f1	f2
max	10.00000	90.000000

```
numdata.mean()
```

```
f1     5.50
f2    51.25
dtype: float64
```

```
tmpdata4 = numdata.fillna( numdata.mean(), inplace=False )
tmpdata4
```

	f1	f2
0	1	10.00
1	2	51.25
2	3	20.00
3	4	30.00
4	5	51.25
5	6	50.00
6	7	60.00
7	8	70.00
8	9	80.00
9	10	90.00

```
missdata.groupby('f3')['f2'].mean()
```

```
f3
A    20.0
B    60.0
C    85.0
Name: f2, dtype: float64
```

```
missdata.groupby('f3')['f2'].transform('mean')
```

```
0    20.0
1    20.0
2    20.0
```

```
3    20.0
4    60.0
5    60.0
6    60.0
7    60.0
8    85.0
9    85.0
Name: f2, dtype: float64
```

```
tmpdata5 = numdata.copy()
tmpdata5['f2'].fillna( missdata.groupby('f3')['f2'].transform('mean'), inplace=True)
tmpdata5
```

```
/var/folders/n2/jbh_0_091bx8qgz7j87t2qwc0000gp/T/ipykernel_25894/622840210.py:2: FutureWarning: A
The behavior will change in pandas 3.0. This inplace method will never work because the intermedi

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value},
```

```
tmpdata5['f2'].fillna( missdata.groupby('f3')['f2'].transform('mean'),inplace=True)
```

	f1	f2
0	1	10.0
1	2	20.0
2	3	20.0
3	4	30.0
4	5	60.0
5	6	50.0
6	7	60.0
7	8	70.0
8	9	80.0
9	10	90.0

```
missdata_tr = missdata.dropna()
x_tr = missdata_tr[['f1']]
y_tr = missdata_tr['f2']

from sklearn.linear_model import LinearRegression
model = LinearRegression()
```

```
model.fit( x_tr, y_tr )

missdata_ts = missdata [ missdata.isnull().any(axis=1) ]
x_ts = missdata_ts[['f1']]

predicted_values = model.predict( x_ts )
tmpdata6 = missdata.copy()
tmpdata6.loc[ tmpdata6['f2'].isnull(), 'f2'] = predicted_values
tmpdata6
```

	f1	f2	f3
0	1	10.000000	A
1	2	14.191176	A
2	3	20.000000	A
3	4	30.000000	A
4	5	41.985294	B
5	6	50.000000	B
6	7	60.000000	B
7	8	70.000000	B
8	9	80.000000	C
9	10	90.000000	C

```
missdata_num = missdata.copy()
missdata_num['f3']=missdata_num['f3'].map({'A':1,'B':2,'C':3})
```

```
missdata_num
```

	f1	f2	f3
0	1	10.0	1
1	2	NaN	1
2	3	20.0	1
3	4	30.0	1
4	5	NaN	2
5	6	50.0	2
6	7	60.0	2
7	8	70.0	2
8	9	80.0	3
9	10	90.0	3

```
from sklearn.impute import KNNImputer
imputer = KNNImputer(n_neighbors=2)
tmpdata7 = imputer.fit_transform(missdata_num)
```

```
pd.DataFrame( tmpdata7 )
```

	0	1	2
0	1.0	10.0	1.0
1	2.0	15.0	1.0
2	3.0	20.0	1.0
3	4.0	30.0	1.0
4	5.0	40.0	2.0
5	6.0	50.0	2.0
6	7.0	60.0	2.0
7	8.0	70.0	2.0
8	9.0	80.0	3.0
9	10.0	90.0	3.0

```
outdict = {'A': [10, 0.02, 0.3, 40, 50, 60, 712, 80, 90, 1003],
           'B': [0.05, 0.00015, 25, 35, 45, 205, 65, 75, 85, 3905]}
outdata = pd.DataFrame( outdict )

Q1 = outdata.quantile(0.25)
Q3 = outdata.quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

((outdata < lower_bound) | (outdata > upper_bound))
```

	A	B
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
5	False	True
6	True	False

	A	B
7	False	False
8	False	False
9	True	True

```
outliers = ((outdata < lower_bound) | (outdata > upper_bound)).any(axis=1)
outliersdata = outdata[ outliers ]
outliersdata
```

	A	B
5	60.0	205.0
6	712.0	65.0
9	1003.0	3905.0

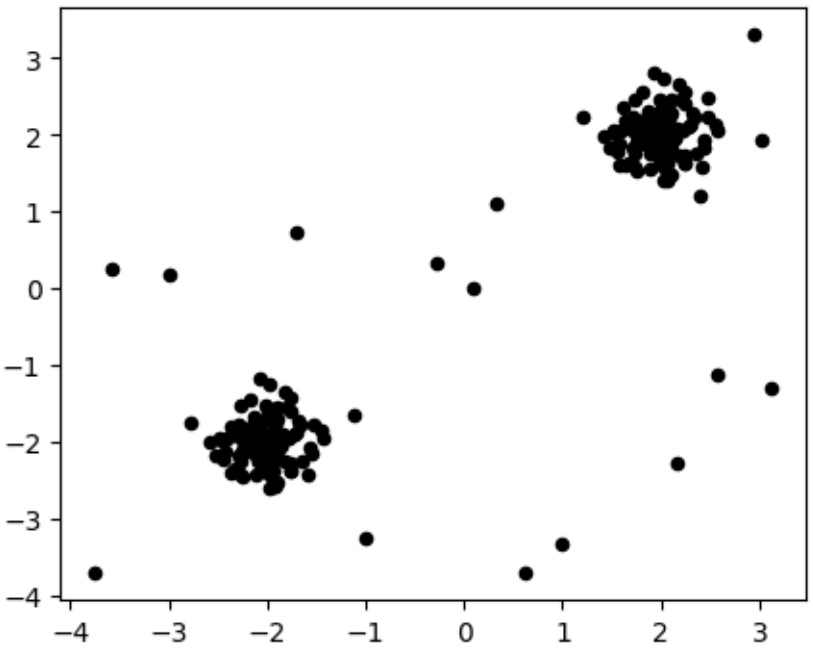
```
standardizeddata = (outdata - outdata.mean()) / outdata.std()
standardizeddata
```

	A	B
0	-0.552206	-0.364647
1	-0.580536	-0.364688
2	-0.579741	-0.344154
3	-0.467047	-0.335940
4	-0.438661	-0.327727
5	-0.410274	-0.196309
6	1.440519	-0.311300
7	-0.353501	-0.303086
8	-0.325115	-0.294872
9	2.266563	2.842723

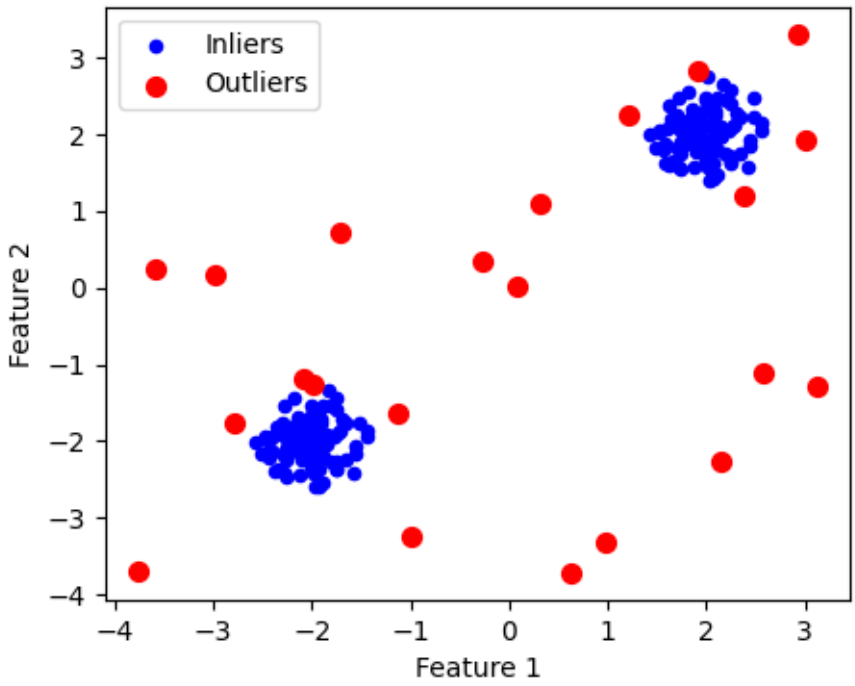
```
outliers2 = ((standardizeddata < -3) | (standardizeddata > 3)).any(axis=1)
outliersdata2 = outdata[ outliers2 ]
outliersdata2
```

	A	B

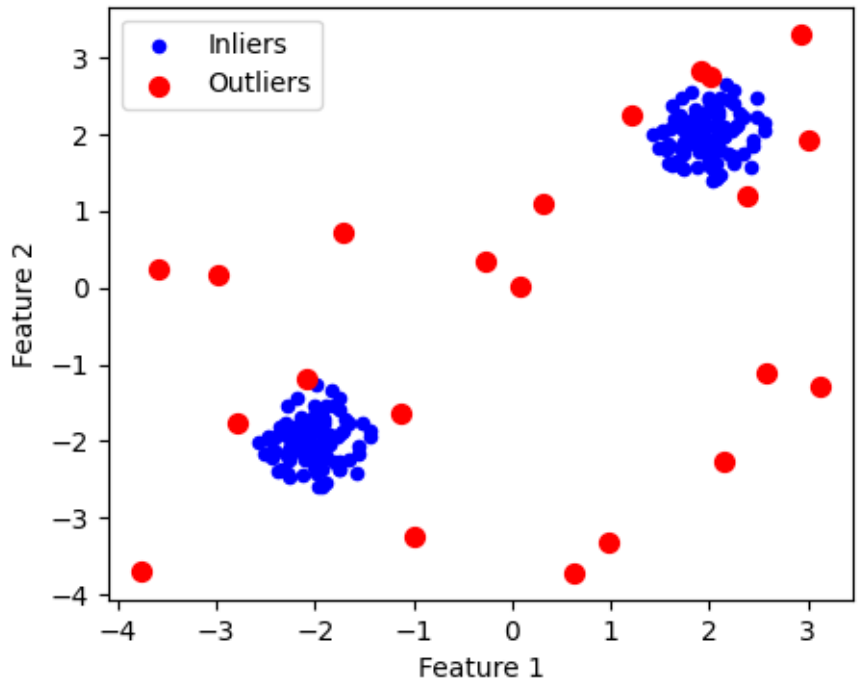
```
import matplotlib.pyplot as plt
np.random.seed(42)
X_inliers = 0.3 * np.random.randn(100, 2)
X_outliers = np.random.uniform(low=-4, high=4, size=(20, 2))
X = np.r_[X_inliers + 2, X_inliers - 2, X_outliers]

plt.figure(figsize=(5, 4))
plt.scatter(X[:, 0], X[:, 1], color='k', s=20)
```




```
from sklearn.neighbors import LocalOutlierFactor
clf = LocalOutlierFactor(n_neighbors=20, contamination=0.1)
y_pred = clf.fit_predict(X) # 1: inlier, -1: outlier
outlier_mask = y_pred == -1

plt.figure(figsize=(5, 4))
plt.scatter(X[:, 0], X[:, 1], color='b', s=20, label='Inliers')
plt.scatter(X[outlier_mask, 0], X[outlier_mask, 1], color='r', s=50,label='Outliers')
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.legend()
```




```
from sklearn.ensemble import IsolationForest
clf2 = IsolationForest(contamination=0.1)
# contamination : 이상치 비율
# n_estimators : 나무의 갯수 (defalut 100)
# max_features : 각 나무별 특성변수의 갯수(default 1)
clf2.fit( X )
y_pred2 = clf2.predict( X ) # 1: inlier, -1: outlier
outlier_mask2 = y_pred2 == -1

plt.figure(figsize=(5, 4))
plt.scatter(X[:, 0], X[:, 1], color='b', s=20, label='Inliers')
plt.scatter(X[outlier_mask2, 0], X[outlier_mask2, 1], color='r', s=50,label='Outliers')
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.legend()
```




```
clf2.score_samples(X)
```

```
array([-0.39567479, -0.43973362, -0.38576701, -0.4625295 , -0.39169087,
       -0.39353101, -0.44632054, -0.4539587 , -0.39410011, -0.42823486,
       -0.43802322, -0.4231585 , -0.38600184, -0.40324911, -0.38778406,
       -0.46849881, -0.4075734 , -0.43420431, -0.45140279, -0.4113938 ,
       -0.40036102, -0.38335266, -0.42666089, -0.41104579, -0.44476313,
       -0.38744281, -0.39942914, -0.44257912, -0.38938554, -0.40470075,
       -0.39006555, -0.42881353, -0.44275864, -0.40154039, -0.39729665,
       -0.42434279, -0.42743206, -0.57699503, -0.38628797, -0.45454533,
       -0.3864026 , -0.44513991, -0.39598029, -0.41231495, -0.39270763,
       -0.40568073, -0.39005843, -0.43210962, -0.38601781, -0.38485125,
       -0.41991455, -0.40181793, -0.38788591, -0.48400217, -0.38538727,
       -0.47693547, -0.50815903, -0.39115267, -0.40423505, -0.43216634,
       -0.4254748 , -0.48475901, -0.4890737 , -0.40357175, -0.39439224,
       -0.42406868, -0.40171635, -0.44870204, -0.38761922, -0.43170548,
       -0.42364173, -0.43593615, -0.39560581, -0.4391784 , -0.39543452,
       -0.38550106, -0.38910014, -0.39558501, -0.48693408, -0.41865632,
       -0.40964165, -0.43730378, -0.41716448, -0.47893783, -0.39791987,
       -0.40492088, -0.38431516, -0.39544321, -0.42208103, -0.53522817,
       -0.41839783, -0.40171635, -0.39362168, -0.39333773, -0.44128807,
       -0.40208128, -0.40907841, -0.39096216, -0.38929625, -0.40645206,
       -0.38953796, -0.43147334, -0.38691831, -0.45292849, -0.39365031,
       -0.40064735, -0.46513506, -0.48209563, -0.39635657, -0.44569517,
```

```
-0.4496465 , -0.43243444, -0.39406682, -0.40625773, -0.39826724,
-0.46462545, -0.40782229, -0.42572527, -0.46599338, -0.42852596,
-0.40082304, -0.38971614, -0.4628486 , -0.4219109 , -0.46365237,
-0.39237271, -0.40320941, -0.42432754, -0.40309339, -0.40204058,
-0.39044555, -0.43501511, -0.4324525 , -0.40503508, -0.40259674,
-0.42956023, -0.42799201, -0.56257644, -0.38875652, -0.47585014,
-0.3835507 , -0.4556408 , -0.406217  , -0.40385118, -0.39458774,
-0.40122018, -0.40401747, -0.4443972 , -0.38320086, -0.3834212 ,
-0.44945534, -0.4060406 , -0.38476589, -0.46817324, -0.38466101,
-0.47610349, -0.49275615, -0.38344594, -0.40974845, -0.42395885,
-0.42189161, -0.49261883, -0.48518689, -0.40901991, -0.39452123,
-0.44785724, -0.40375287, -0.45749968, -0.40496115, -0.4260838 ,
-0.41810669, -0.44560803, -0.38806155, -0.45523273, -0.38817405,
-0.38177641, -0.39724261, -0.40279204, -0.46677259, -0.42162856,
-0.4086809 , -0.43961327, -0.40404401, -0.45862325, -0.40473907,
-0.41509113, -0.38081908, -0.39036169, -0.42441162, -0.52130968,
-0.41557892, -0.40347146, -0.39318444, -0.38921027, -0.46342999,
-0.39756969, -0.41357841, -0.38429305, -0.40297782, -0.40881293,
-0.60259641, -0.44411547, -0.57052254, -0.50247903, -0.73212838,
-0.64786135, -0.55623141, -0.45375541, -0.68754218, -0.67833274,
-0.70698798, -0.63422967, -0.64031155, -0.78183101, -0.65477657,
-0.67178443, -0.67084008, -0.68580391, -0.71446705, -0.64861428])
```

2 빅데이터와 금융자료 분석 기말대체과제

20249132 김형환

2.1 Question 1

1. prob1_bank.csv 자료는 어느 포르투갈 은행의 정기예금 프로모션 전화 데이터이다. 이 데이터는 고객의 특징을 나타내는 특성 변수들과 고객이 정기예금에 가입했는지 여부를 나타내는 목표 변수로 구성되어 있다.

<특성변수>
age : 나이
job : 직업의 형태
marital : 결혼 상태
education : 학력
default : 신용 불이행 여부
balance : 은행 잔고
housing : 부동산 대출 여부
loan : 개인 대출 여부
contact : 연락 수단
month : 마지막으로 연락한 달

<목표변수>
y : 고객이 정기 예금에 가입했는지 여부

2.1.1 (1)

주어진 자료 중 범주형 변수 각각에 대해 적절한 전처리를 선택하고 진행하여라.

```
import numpy as np
import pandas as pd

bank = pd.read_csv('data/prob1_bank.csv')
bank.info() # 전체 11개 칼럼에 null값은 없으며, 범주형 9개 및 숫자형 2개
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4521 entries, 0 to 4520
Data columns (total 11 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
```

```
 0   age        4521 non-null   int64
 1   job        4521 non-null   object
 2   marital    4521 non-null   object
 3   education  4521 non-null   object
 4   default    4521 non-null   object
 5   balance    4521 non-null   int64
 6   housing    4521 non-null   object
 7   loan       4521 non-null   object
 8   contact    4521 non-null   object
 9   month      4521 non-null   object
 10  y          4521 non-null   object
dtypes: int64(2), object(9)
memory usage: 388.6+ KB
```

```
bank.select_dtypes(include='object').nunique()
# 범주형 9개 중, 목적변수를 포함하여 4개는 '여부'에 대한 이진변수이며 나머지는 3~12개의 고유값
```

```
job          12
marital       3
education     4
default       2
housing       2
loan          2
contact       3
month        12
y             2
dtype: int64
```

```
bank['job'].value_counts()
# 직업의 경우, 12개의 범주로 구성됨. 특별히 한 값에 치중되는 모습도 보이지 않고,
# 순서가 없으므로 원-핫 인코딩으로 처리 예정
```

```
job
management       969
blue-collar      946
technician       768
admin.           478
services         417
retired          230
```

```
self-employed    183
entrepreneur     168
unemployed       128
housemaid        112
student           84
unknown           38
Name: count, dtype: int64
```

```
bank['marital'].value_counts()
# 결혼 여부는 싱글/결혼/이혼 3진변수임. 순서가 없으므로 원-핫 인코딩으로 처리 예정
```

```
marital
married     2797
single      1196
divorced     528
Name: count, dtype: int64
```

```
bank['education'].value_counts()
# 학력에 대한 내용은 primary - secondary - tertiary 순이므로 1~3 라벨인코딩 처리 예정
# unknown은 학력수준이 낮을 가능성이 높을 것으로 추정, 0으로 처리하여 하나의 응답값으로 간주함.
```

```
education
secondary    2306
tertiary     1350
primary       678
unknown       187
Name: count, dtype: int64
```

```
bank['contact'].value_counts()
# 연락방법에 대한 내용은 telephone - cellular 순으로 연락이 편리하므로 1~2 라벨인코딩 처리 예정
# unknown은 연락 편의성이 가장 떨어질 것으로 보임. 0으로 처리하여 라벨인코딩 처리 예정
```

```
contact
cellular     2896
unknown      1324
telephone     301
Name: count, dtype: int64
```

```
bank['month'].value_counts()
# 달에 대한 내용으로, 1~12개월 순서에 따른 라벨인코딩 처리 예정
```

```
month
may    1398
jul     706
aug     633
jun     531
nov     389
apr     293
feb     222
jan     148
oct      80
sep      52
mar      49
dec      20
Name: count, dtype: int64
```

```
# 범주형 변수 처리

# 1. job, marital -> One-Hot Encoding
bank = pd.get_dummies(bank, columns=['job', 'marital'], drop_first=True)

# 2. education, contact, month > Label Encoding
edu_map = {'unknown': 0,'primary': 1,'secondary': 2,'tertiary': 3}
bank['education'] = bank['education'].map(edu_map)

contact_map = {'unknown': 0,'telephone': 1,'cellular': 2}
bank['contact'] = bank['contact'].map(contact_map)

month_order = ['jan', 'feb', 'mar', 'apr', 'may', 'jun',
               'jul', 'aug', 'sep', 'oct', 'nov', 'dec']
month_map = {month: i for i, month in enumerate(month_order)}
bank['month'] = bank['month'].map(month_map)

# 3. default, housing, loan, y > Label Encoding (binary, yes=1 / no=0)
binary_cols = ['default', 'housing', 'loan', 'y']
for col in binary_cols: bank[col] = bank[col].map({'yes': 1, 'no': 0})
```

```
print(bank.info())
```

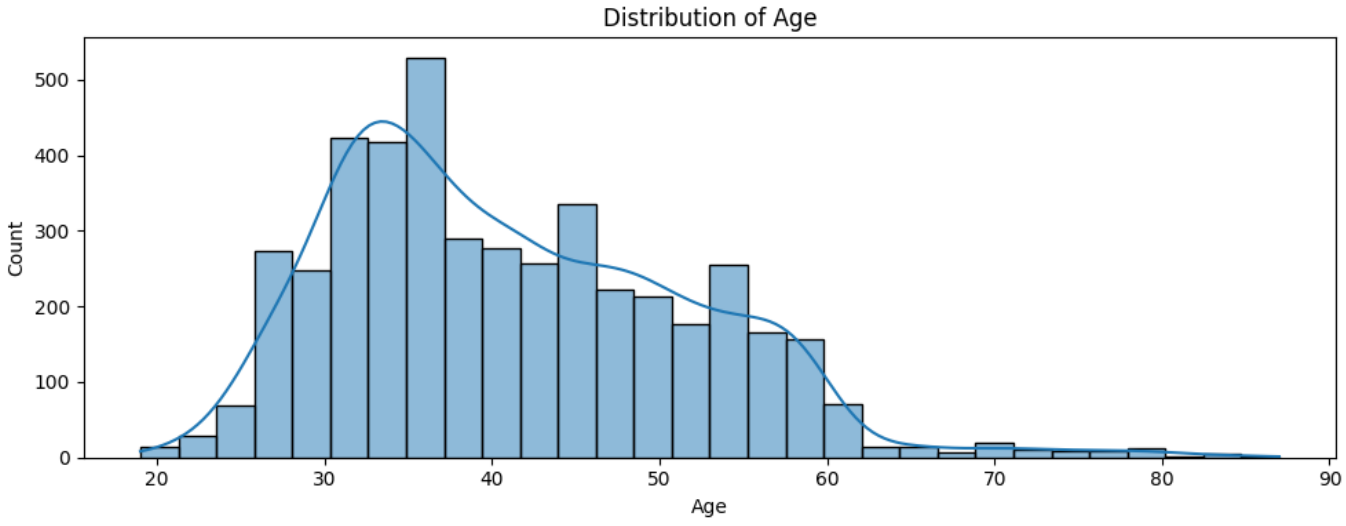
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4521 entries, 0 to 4520
Data columns (total 22 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   age                 4521 non-null   int64
 1   education           4521 non-null   int64
 2   default             4521 non-null   int64
 3   balance             4521 non-null   int64
 4   housing             4521 non-null   int64
 5   loan                4521 non-null   int64
 6   contact             4521 non-null   int64
 7   month               4521 non-null   int64
 8   y                   4521 non-null   int64
 9   job_blue-collar     4521 non-null   bool
 10  job_entrepreneur    4521 non-null   bool
 11  job_housemaid       4521 non-null   bool
 12  job_management      4521 non-null   bool
 13  job_retired         4521 non-null   bool
 14  job_self-employed   4521 non-null   bool
 15  job_services        4521 non-null   bool
 16  job_student         4521 non-null   bool
 17  job_technician      4521 non-null   bool
 18  job_unemployed      4521 non-null   bool
 19  job_unknown         4521 non-null   bool
 20  marital_married     4521 non-null   bool
 21  marital_single      4521 non-null   bool
dtypes: bool(13), int64(9)
memory usage: 375.4 KB
None
```

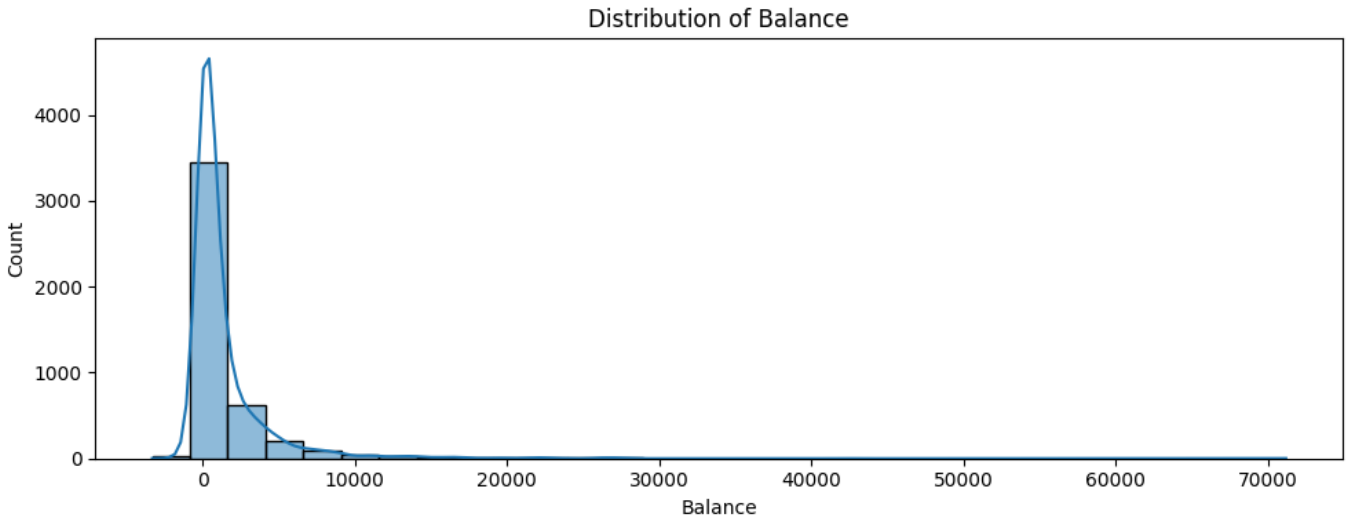
2.1.2 (2)

주어진 자료 중 수치형 변수 각각에 대해 적절한 전처리를 선택하고 진행하여라.

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# 1. age 분포
plt.figure(figsize=(10, 4))
sns.histplot(bank['age'], bins=30, kde=True)
plt.title("Distribution of Age")
plt.xlabel("Age")
plt.ylabel("Count")
plt.tight_layout()
plt.show()

# 2. balance 분포
plt.figure(figsize=(10, 4))
sns.histplot(bank['balance'], bins=30, kde=True)
plt.title("Distribution of Balance")
plt.xlabel("Balance")
plt.ylabel("Count")
plt.tight_layout()
plt.show()
```





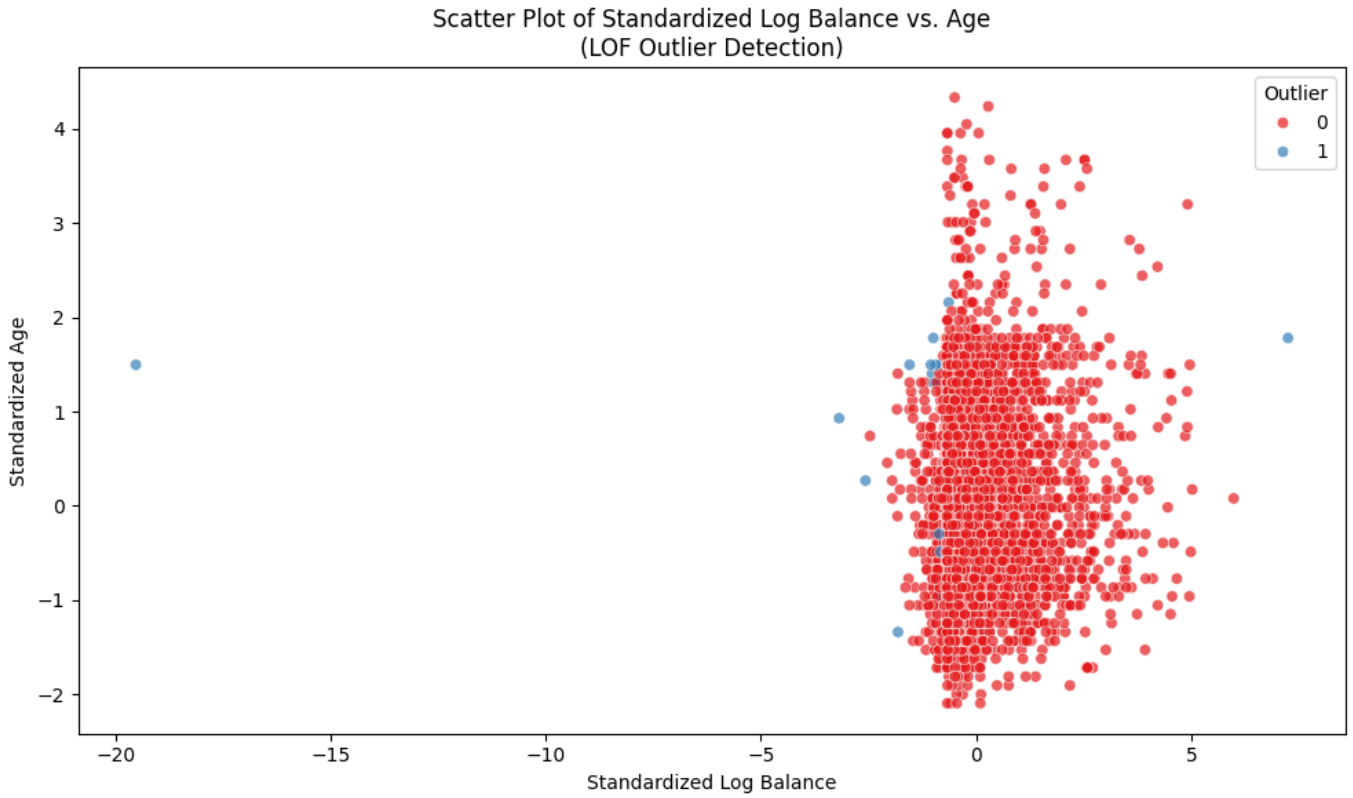

```
bank['age'].describe()
# age는 오른쪽 skew가 미세하게 있는 정규분포와 가까운 형태. 표준화만 진행
```

```
count    4521.000000
mean       41.170095
std        10.576211
min        19.000000
25%        33.000000
50%        39.000000
75%        49.000000
max        87.000000
Name: age, dtype: float64
```

```
bank['balance'].describe()
# balance는 음수도 존재하고, 값이 매우 극단적으로 치우쳐져 있는 형태. log변환 및 표준화 진행
# 이후 LOF 방식으로 두 수치형변수에 대한 이상치 탐지, 약 1% 수준의 이상치 제거 예정
```

```
count     4521.000000
mean      1422.657819
std       3009.638142
min      -3313.000000
25%         69.000000
50%        444.000000
75%       1480.000000
max      71188.000000
Name: balance, dtype: float64
```

```python
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import LocalOutlierFactor
import numpy as np

# 1. balance 로그변환 (음수 대비)
min_bal = bank['balance'].min()
bank['log_balance'] = np.log1p(bank['balance'] - min_bal + 1)

# 2. balance + age 표준화
scaler = StandardScaler()
bank[['std_log_balance', 'std_age']] = scaler.fit_transform(bank[['log_balance', 'age']])

# 3. LOF 이상치 탐지 (다변량: log_balance + age)
# 적절한 파라미터 조정으로 balance의 최소, 최대값을 효과적으로 제거
X_scaled = bank[['std_log_balance', 'std_age']]
lof = LocalOutlierFactor(n_neighbors=30, contamination=0.01)
bank['is_outlier'] = (lof.fit_predict(X_scaled) == -1).astype(int)

# 4. 이상치 시각화 (scatter plot)
plt.figure(figsize=(10, 6))
sns.scatterplot(data=bank, x='std_log_balance', y='std_age',
                hue='is_outlier', palette='Set1', alpha=0.7)
plt.title("Scatter Plot of Standardized Log Balance vs. Age\n(LOF Outlier Detection)")
plt.xlabel("Standardized Log Balance")
plt.ylabel("Standardized Age")
plt.legend(title="Outlier")
plt.tight_layout()
plt.show()
```



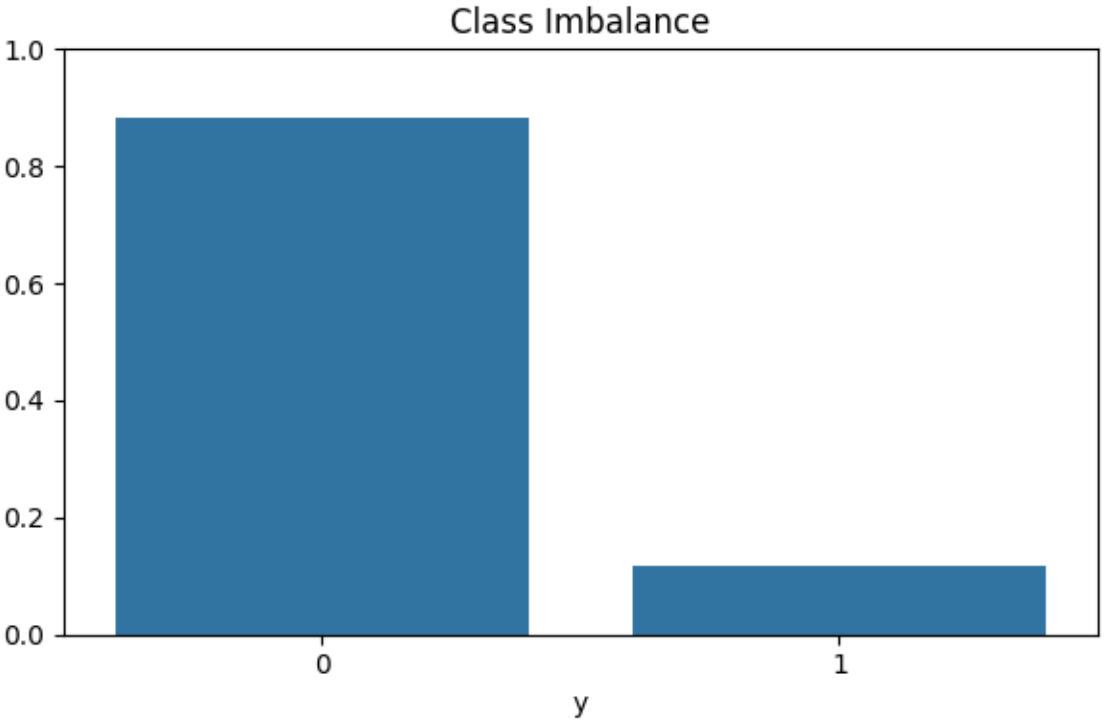

```
# 이상치 1%가 제거된 것을 확인할 수 있음
bank_clean = bank[bank['is_outlier'] == 0].copy()
bank_clean
```

	age	education	default	balance	housing	loan	contact	month	y	job_blue-collar	...	job_stude
0	30	1	0	1787	0	0	2	9	0	False	...	False
1	33	2	0	4789	1	1	2	4	0	False	...	False
2	35	3	0	1350	1	0	2	3	0	False	...	False
3	30	3	0	1476	1	1	0	5	0	False	...	False
4	59	2	0	0	1	0	0	4	0	True	...	False
...	...	...	...	...	...	...	...	...	...	...	...	...
4515	32	2	0	473	1	0	2	6	0	False	...	False
4516	33	2	0	-333	1	0	2	6	0	False	...	False
4518	57	2	0	295	0	0	2	7	0	False	...	False
4519	28	2	0	1137	0	0	2	1	0	True	...	False
4520	44	3	0	1136	1	1	2	3	0	False	...	False

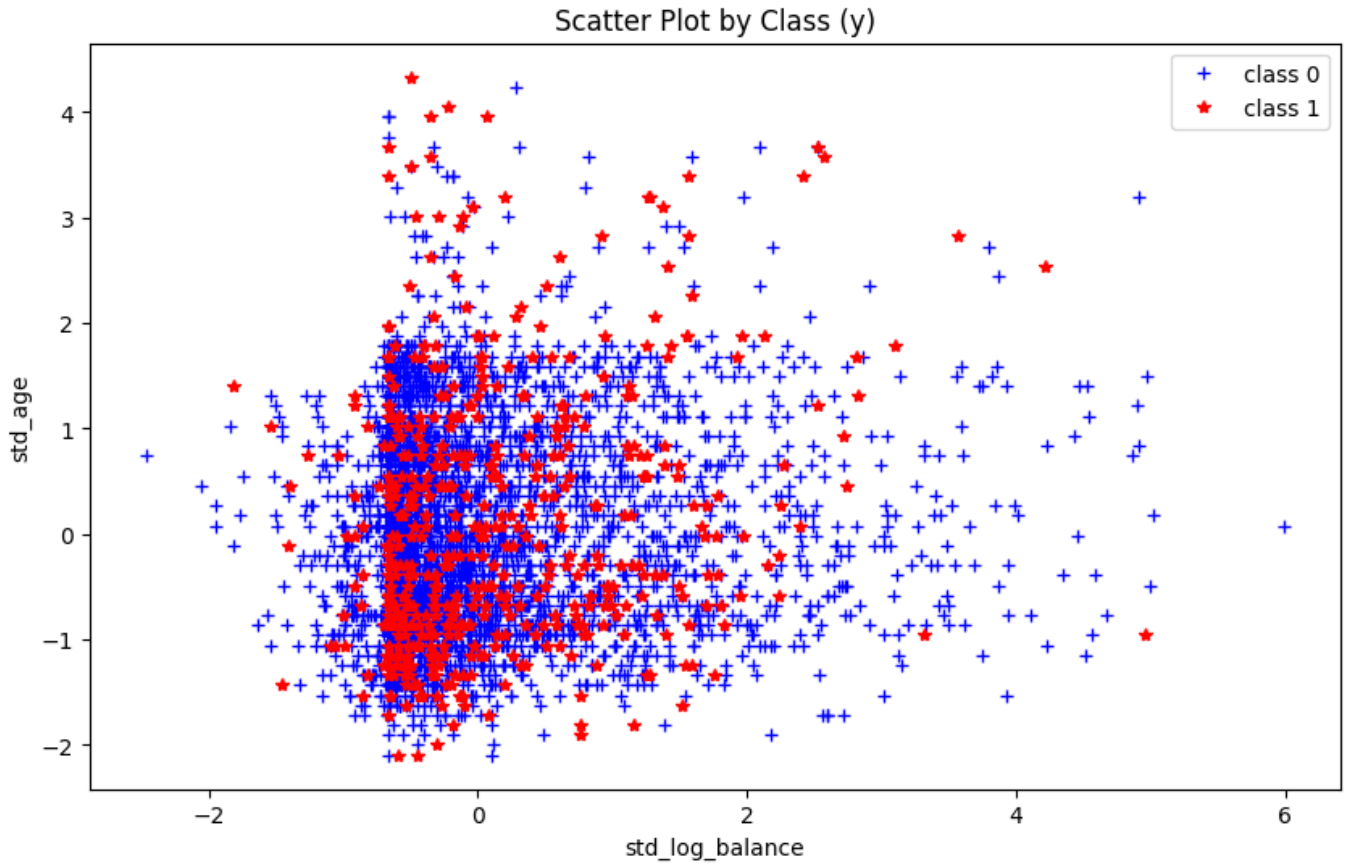
2.1.3 (3)

주어진 자료에 클래스 불균형이 있는지 확인한 뒤, 이에 대한 적절한 전처리 방법을 선택하여 진행하여라.

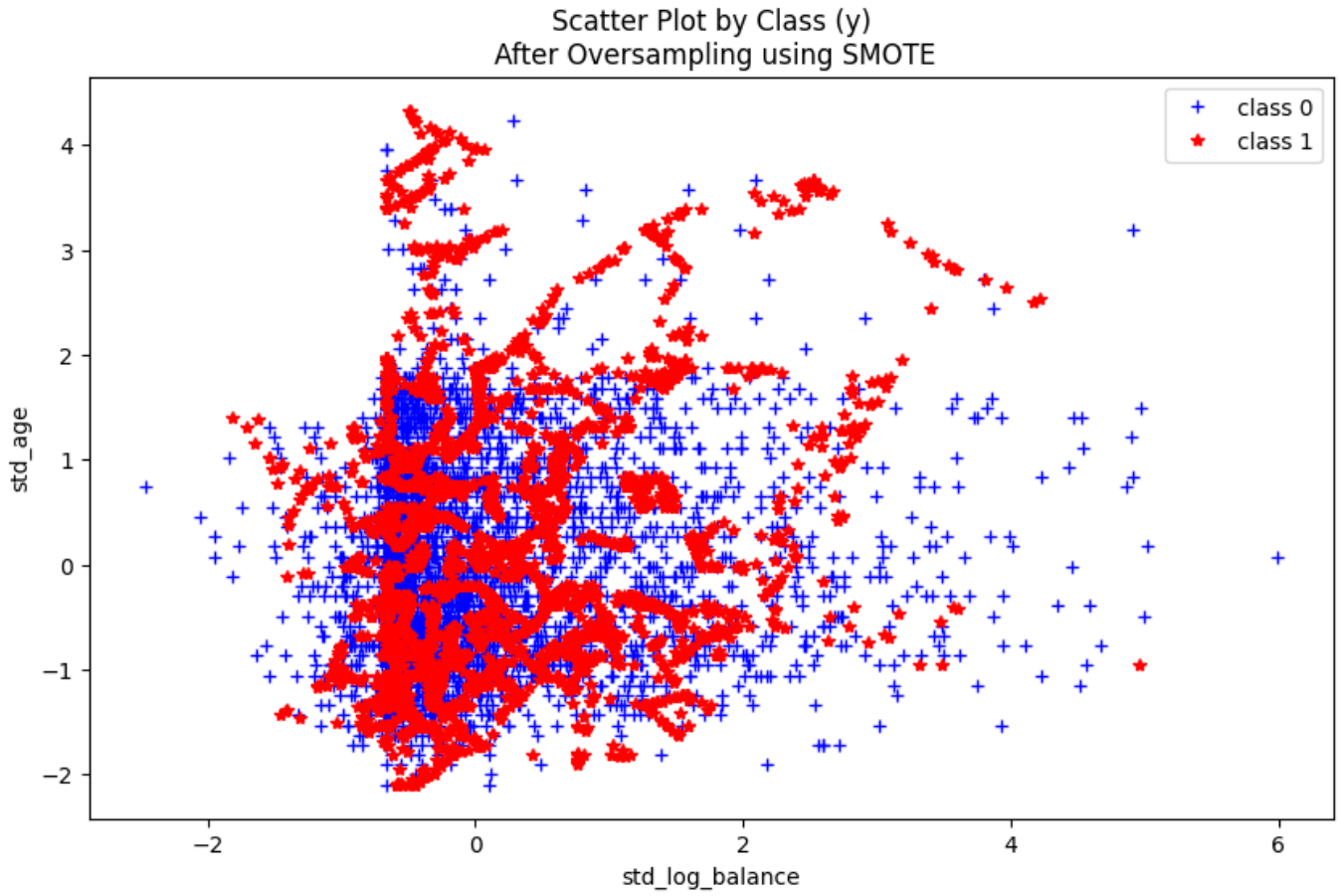
```
# 약 9:1로 0(No)의 비율이 압도적으로 많음. 클래스불균형 존재

class_counts = bank_clean['y'].value_counts(normalize=True)
plt.figure(figsize=(6, 4))
sns.barplot(x=class_counts.index, y=class_counts.values, legend=False)
plt.title("Class Imbalance")
plt.xlabel("y")
plt.ylim(0, 1)
plt.tight_layout()
plt.show()
```




```
# 수치형 변수의 scatter plot상으로 특정 수치형 변수에 따른 치우침은 관측되지 않음.

X = bank_clean[['std_log_balance','std_age']]
y = bank_clean['y']
plt.figure(figsize=(10, 6))
plt.plot(X.loc[y == 0, 'std_log_balance'], X.loc[y == 0, 'std_age'], 'b+', label="class 0")
plt.plot(X.loc[y == 1, 'std_log_balance'], X.loc[y == 1, 'std_age'], 'r*', label="class 1")
plt.legend()
plt.xlabel("std_log_balance")
plt.ylabel("std_age")
plt.title("Scatter Plot by Class (y)")
plt.show()
```




```
# 표본의 수가 적은 편이므로 oversampling 진행
# 데이터의 구조가 그다지 복잡하지 않아 SMOTE 알고리즘을 적용하여 처리
from imblearn.over_sampling import SMOTE

oversample1 = SMOTE()
OX, Oy = oversample1.fit_resample(X, y)
plt.figure(figsize=(10, 6))
plt.plot(OX.loc[Oy == 0, 'std_log_balance'], OX.loc[Oy == 0, 'std_age'], 'b+', label="class 0"
plt.plot(OX.loc[Oy == 1, 'std_log_balance'], OX.loc[Oy == 1, 'std_age'], 'r*', label="class 1"
plt.legend()
plt.xlabel("std_log_balance")
plt.ylabel("std_age")
plt.title("Scatter Plot by Class (y)\n After Oversampling using SMOTE")
plt.show()
```



2.2 Question 2

2. prob2_card.csv 자료는 어느 신용카드 회사의 고객 데이터로, 신용카드 사용 형태를 나타내는 여러 특성 변수들로 구성되어 있다.

CUST_ID : 신용카드 사용자 ID
BALANCE : 구매 계좌 잔액
BALANCE_FREQUENCY : 구매 계좌 잔액이 업데이트 되는 빈도 지수로, 0(자주 업데이트 되지 않음)~1(자주 업데이트 됨) 사이의 값을 가짐.
PURCHASES : 구매 계좌로부터의 구매액
PURCHASES_FREQUENCY : 구매 빈도 지수로, 0(자주 구매하지 않음)~1(자주 구매함) 사이의 값을 가짐.
PURCHASES_TRX : 구매 거래 건수

2.2.1 (1)

주어진 자료에 K평균 Clustering 알고리즘을 적용하여, 적절한 군집을 생성하여라.

```
df = pd.read_csv('data/prob2_card.csv')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8950 entries, 0 to 8949
```

```
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   CUST_ID              8950 non-null   object
 1   BALANCE              8950 non-null   float64
 2   BALANCE_FREQUENCY    8950 non-null   float64
 3   PURCHASES            8950 non-null   float64
 4   PURCHASES_FREQUENCY  8950 non-null   float64
 5   PURCHASES_TRX        8950 non-null   int64
dtypes: float64(4), int64(1), object(1)
memory usage: 419.7+ KB
```

```
df.describe()
```

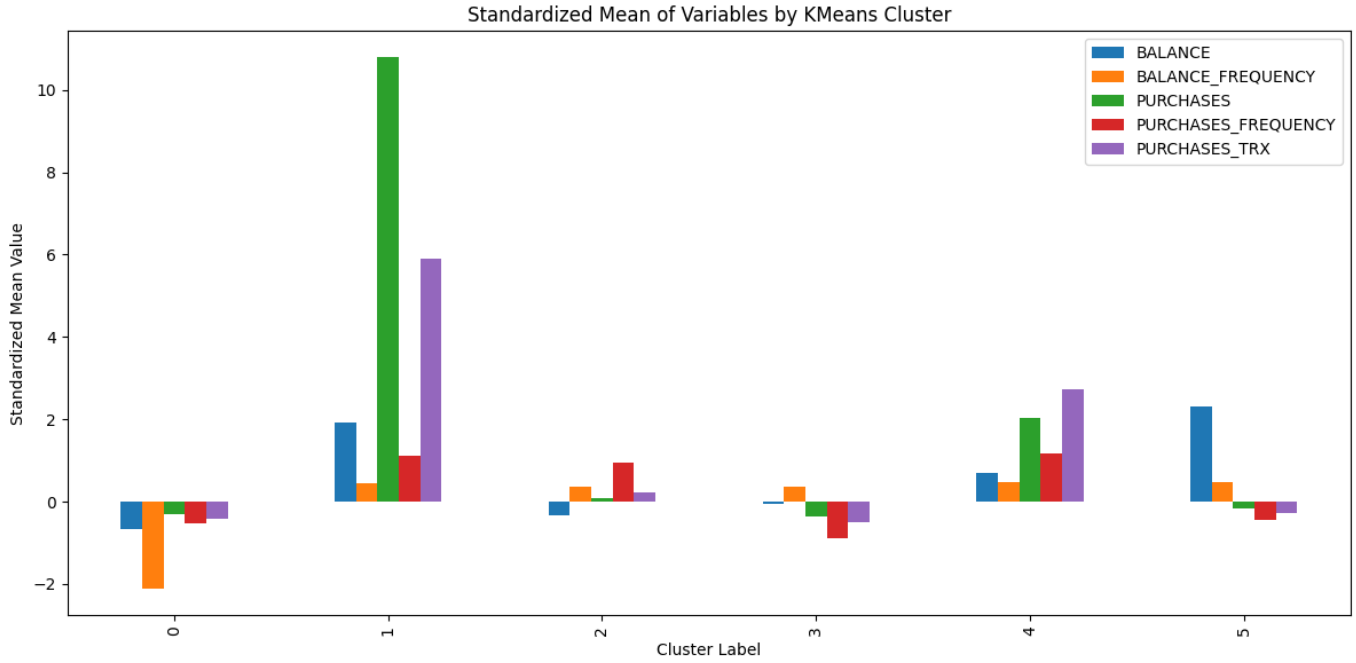
	BALANCE	BALANCE_FREQUENCY	PURCHASES	PURCHASES_FREQUENCY	PURCHASE
count	8950.000000	8950.000000	8950.000000	8950.000000	8950.000000
mean	1564.474828	0.877271	1003.204834	0.490351	14.709832
std	2081.531879	0.236904	2136.634782	0.401371	24.857649
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	128.281915	0.888889	39.635000	0.083333	1.000000
50%	873.385231	1.000000	361.280000	0.500000	7.000000
75%	2054.140036	1.000000	1110.130000	0.916667	17.000000
max	19043.138560	1.000000	49039.570000	1.000000	358.000000

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# CUST_ID는 클러스터링에 사용하지 않으므로 제거
X = df.drop(columns=["CUST_ID"])

# 표준화
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
scaled_df = pd.DataFrame(X_scaled, columns=X.columns)

# 최적 K 탐색
inertia_list = []
silhouette_list = []
K_range = range(2, 11)
```

```
for k in K_range:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    labels = kmeans.fit_predict(X_scaled)
    inertia_list.append(kmeans.inertia_)
    silhouette_list.append(silhouette_score(X_scaled, labels))

best_k = K_range[silhouette_list.index(max(silhouette_list))]

# K평균 군집화 결과 : 6개의 군집으로 분류되었으며, 갯수는 32개~3200개로 천차만별
kmeans_final = KMeans(n_clusters=best_k, random_state=42, n_init=10)
scaled_df["KMeans_Label"] = kmeans_final.fit_predict(X_scaled)

summary_table_kmean = scaled_df.groupby("KMeans_Label").mean()
summary_table_kmean["Count"] = scaled_df.groupby("KMeans_Label").size()
summary_table_kmean
```

KMeans_Label	BALANCE	BALANCE_FREQUENCY	PURCHASES	PURCHASES_FREQUENCY	PU
0	-0.679901	-2.105347	-0.305189	-0.528487	-0.4
1	1.906921	0.444930	10.782703	1.119309	5.89
2	-0.331950	0.368356	0.087095	0.960328	0.22
3	-0.068576	0.366930	-0.369541	-0.895659	-0.5
4	0.696282	0.475045	2.042474	1.179006	2.71
5	2.320551	0.483046	-0.159336	-0.433551	-0.2

```
# K평균 군집화 시각화
summary_table_kmean.drop(columns="Count").plot.bar(figsize=(12, 6))
plt.title("Standardized Mean of Variables by KMeans Cluster")
plt.xlabel("Cluster Label")
plt.ylabel("Standardized Mean Value")
plt.tight_layout()
```



2.2.2 (2)

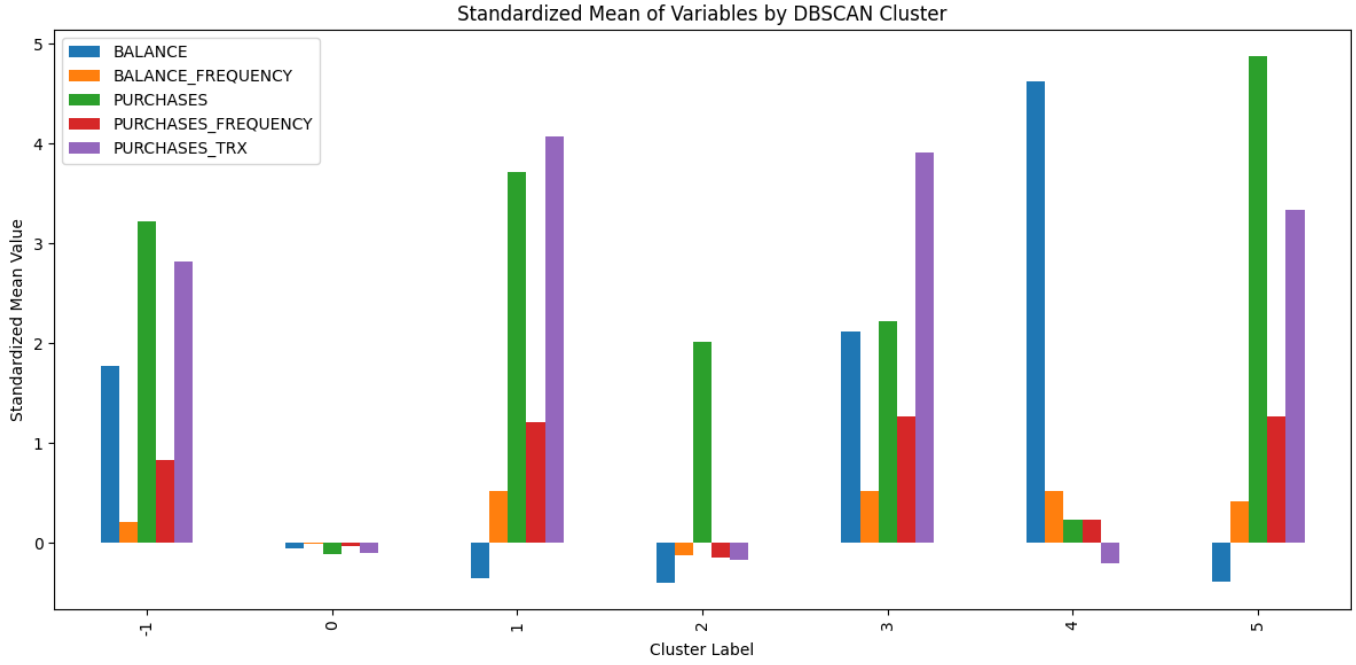
주어진 자료에 DBSCAN Clustering 알고리즘을 적용하여, 적절한 군집을 생성하여라.

```
from sklearn.cluster import DBSCAN

# DBSCAN 분류 : eps 및 min_samples는 여러번 반복을 통해 최적의 조합을 도출하였음.
# 기준 : 분류가 너무 많거나 적지 않도록(3~6개), 너무 숫자가 적은 분류가 없도록
dbscan = DBSCAN(eps=0.6, min_samples=4)
labels = dbscan.fit_predict(X_scaled)

# 라벨을 데이터프레임에 추가
scaled_df["DBSCAN_Label"] = labels

# DBSCAN 군집화 결과 : 잡음(-1)이 약 2% 포함되어있으며, 총 5개의 군집으로 분류(177개~6200개)
summary_table_db = scaled_df.drop(columns="KMeans_Label").groupby("DBSCAN_Label").mean()
summary_table_db["Count"] = scaled_df.groupby("DBSCAN_Label").size()
summary_table_db
```

DBSCAN_Label	BALANCE	BALANCE_FREQUENCY	PURCHASES	PURCHASES_FREQUENCY	PU
-1	1.776034	0.211613	3.221764	0.832535	2.8
0	-0.056781	-0.007807	-0.107234	-0.028259	-0.
1	-0.355658	0.518084	3.714917	1.207553	4.0

DBSCAN_Label	BALANCE	BALANCE_FREQUENCY	PURCHASES	PURCHASES_FREQUENCY	PU
2	-0.397981	-0.121515	2.012334	-0.148985	-0.
3	2.124764	0.518084	2.221881	1.269843	3.9
4	4.623668	0.518084	0.235086	0.231676	-0.
5	-0.384296	0.422144	4.876114	1.269843	3.3

```
summary_table_db.drop(columns="Count").plot.bar(figsize=(12, 6))
plt.title("Standardized Mean of Variables by DBSCAN Cluster")
plt.xlabel("Cluster Label")
plt.ylabel("Standardized Mean Value")
plt.tight_layout()
```



2.2.3 (3)

(1)과 (2)의 두 군집 분석 결과를 비교하고, 더 타당한 모델을 선택하여라.

K평균 vs DBSCAN 군집화 결과 비교

항목	K평균	DBSCAN
클러스터 수	6개	6개 (-1 포함)
클러스터 구성	최소 32, 최대 3253명	최소 4, 최대 8657
이상치 처리	없음	-1로 261명 처리
군집별 특성	분류기준 명확하	분류기준 모호, 대부분 하나로 분류

시각적 해석력	변수별 차이 뚜렷	구분 어려움

결론 : 이 데이터에서는 **K평균이 더 타당한 군집화 방법**

2.2.4 (4)

(3)에서 선택된 최종 모델로 생성한 군집들의 고객 특성을 분석하여라.

K평균 군집화 군집별 특성

0 : 낮은 자산, 낮은 구매액, 낮은 구매횟수 -> 하위 고객군 (약 17.5%)

1 : 높은 자산, 높은 구매액, 높은 구매횟수 -> 부유하고 이용량 많은 VIP 고객군 (약 0.5%)

2 : 평균이하 자산, 평균적인 구매액, 평균이상 구매횟수 -> 일반 고객군 중 상위 이용고객 (약 35%)

3 : 평균적인 자산, 평균이하 구매액, 평균이하 구매횟수 -> 일반 고객군 중 하위 이용고객 (약 33%)

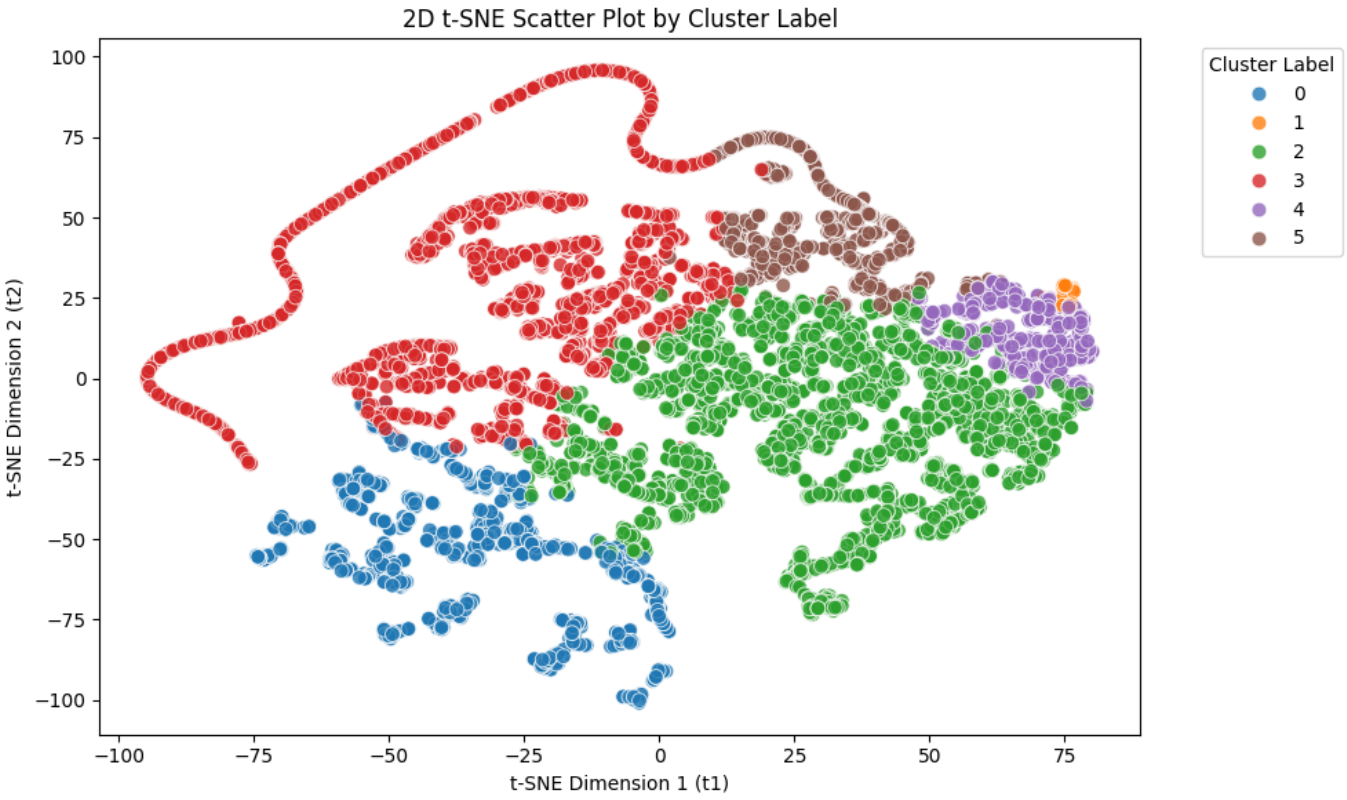
4 : 평균적인 자산, 높은 구매액, 높은 구매횟수 -> 평균적이나 카드 사용량이 많은 우량 고객군 (약 5%)

5 : 높은 자산, 낮은 구매액, 낮은 구매횟수 -> 부유하나 카드를 이용하지 않는 잠재 고객군 (약 9%)

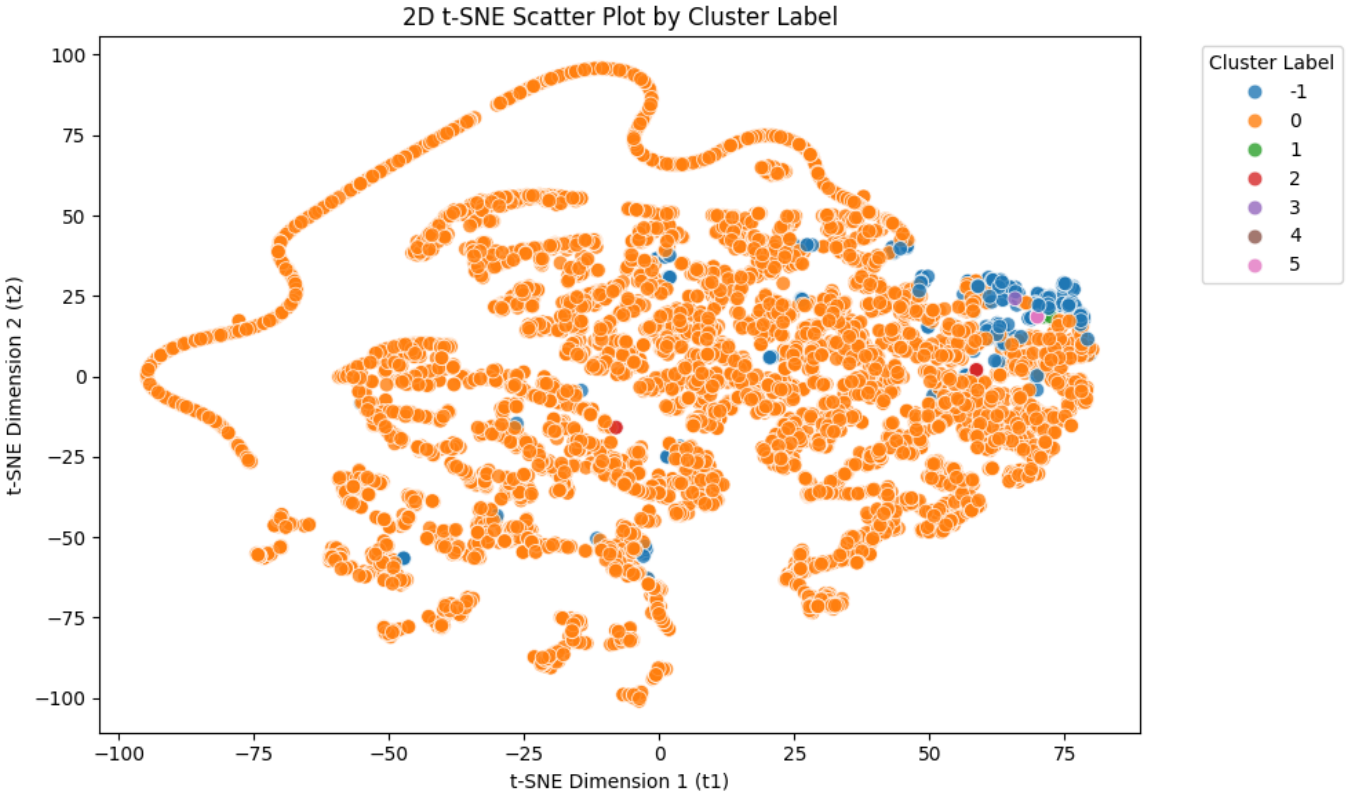
2.2.5 (5)

t-SNE 알고리즘을 적용하여 주어진 자료를 2차원으로 축소하여라. 그 결과를, (3)에서 선택한 모델의 군집 레이블에 따라 점의 색상이 다르게 표현된 2차원 산점도로 시각화하여라.

```
from sklearn.manifold import TSNE
tsne = TSNE( n_components=2 )
df2dim = tsne.fit_transform( X_scaled )
df2dim = pd.DataFrame( df2dim, columns=['t1','t2'] )
df2dim['Labels' ] = scaled_df["KMeans_Label"]

plt.figure(figsize=(10, 6))
sns.scatterplot(data=df2dim, x="t1", y="t2", hue="Labels", palette="tab10", s=60, alpha=0.8)
plt.title("2D t-SNE Scatter Plot by Cluster Label")
plt.xlabel("t-SNE Dimension 1 (t1)")
plt.ylabel("t-SNE Dimension 2 (t2)")
plt.legend(title="Cluster Label", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```




```
# 참고 : DBSCAN 시각화 결과
df2dim['Labels' ] = scaled_df["DBSCAN_Label"]


plt.figure(figsize=(10, 6))
sns.scatterplot(data=df2dim, x="t1", y="t2", hue="Labels", palette="tab10", s=60, alpha=0.8)
plt.title("2D t-SNE Scatter Plot by Cluster Label")
plt.xlabel("t-SNE Dimension 1 (t1)")
plt.ylabel("t-SNE Dimension 2 (t2)")
plt.legend(title="Cluster Label", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```



2.3 Question 3

3. 다음 데이터에서 특성변수인 X는 방의 개수, 목표변수인 Y는 주택 가격을 나타낸다.

ID	X	Y
1	3	1.25
2	1	1.2
3	2	1.3
4	4	1.5
5	?	1.4
6	?	1.3

2.3.1 (1)

XGBoost 알고리즘을 적용하여 트리를 생성한다고 할 때, 첫번째 트리의 첫 마디에서 최적의 분리기준이 무엇인지를 구하여라. 단, 결측이 아닌 4개의 관찰치(ID 1~ID 4)만 이용할 것. 제곱오차 손실함수를 적용하며, 모델 초기값 0는 0.5로 두고, 규제 하이퍼 파라미터 는 0으로 설정할 것. 또한 계산 과정을 상세하게 서술할 것.

1. Gradient 및 Hessian 계산

- 손실함수: $(y_i - \hat{y}_i)^2$
- 예측값: $\hat{y}_i = 0.5$
- Gradient: $g_i = \hat{y}_i - y_i = 0.5 - y_i$
- Hessian: $h_i = 1$ (제곱오차 손실함수는 상수)

ID	y_i	g_i	h_i
1	1.25	-0.75	1
2	1.20	-0.70	1
3	1.30	-0.80	1
4	1.50	-1.00	1

---

2. Split 후보 및 Gain 계산

Gain 공식 ($\lambda = 0$):

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L} + \frac{G_R^2}{H_R} - \frac{(G_L + G_R)^2}{H_L + H_R}\right]$$

Split 1: $X \leq 1.5$

- Left: ID 2 → $G_L = -0.70$, $H_L = 1$
- Right: ID 1, 3, 4 → $G_R = -2.55$, $H_R = 3$

$$\text{Gain}_1 = \frac{1}{2}(0.49 + 2.1675 - 2.640625) = 0.0084$$

Split 2: $X \leq 2.5$

- Left: ID 2, 3 → $G_L = -1.50$, $H_L = 2$
- Right: ID 1, 4 → $G_R = -1.75$, $H_R = 2$

$$\text{Gain}_2 = \frac{1}{2}(1.125 + 1.53125 - 2.640625) = 0.0078$$

Split 3: $X \leq 3.5$

- Left: ID 1, 2, 3 → $G_L = -2.25$, $H_L = 3$

- Right: ID 4 → $G_R = -1.00$, $H_R = 1$

$$\text{Gain}_3 = \frac{1}{2}(1.6875 + 1 - 2.640625) = 0.0234$$

결론

Split 조건	Gain
$X \leq 1.5$	0.0084
$X \leq 2.5$	0.0078
$X \leq 3.5$	**0.0234**

최적 분리 기준: $X \leq 3.5$

2.3.2 (2)

XGBoost 알고리즘을 적용하여 트리를 생성한다고 할 때, 첫번째 트리의 첫 마디에서 X의 값이 결측인 경우는 왼쪽과 오른쪽 자식마디 중 어느 쪽으로 보내야 할지를 결정하여라.

위의 최적 분할 기준 $X \leq 3.5$에 따라, 아래 상황임.

- Left: ID 1, 2, 3 → $G_L = -2.25$, $H_L = 3$
- Right: ID 4 → $G_R = -1.00$, $H_R = 1$

---

1. 결측이 ID 5인 경우 ($Y = 1.4$)

- $g_5 = 0.5 - 1.4 = -0.9$, $h_5 = 1$

Case A: 왼쪽으로 보낼 경우

- $G_L = -3.15$, $H_L = 4$
- $G_R = -1.00$, $H_R = 1$

$$\text{Gain}_{5,\text{left}} = \frac{1}{2}\left(\frac{(-3.15)^2}{4} + \frac{(-1)^2}{1} - \frac{(-4.15)^2}{5}\right) = 0.0181$$

Case B: 오른쪽으로 보낼 경우

- $G_L = -2.25$, $H_L = 3$

- $G_R = -1.9$, $H_R = 2$

$$\text{Gain}_{5,\text{right}} = \frac{1}{2}\left(\frac{(-2.25)^2}{3} + \frac{(-1.9)^2}{2} - \frac{(-4.15)^2}{5}\right) = 0.0240$$

결론: 결측값이 ID 5인 경우, 오른쪽이 더 유리함

---

2. 결측이 ID 6인 경우 ($Y = 1.3$)

- $g_6 = 0.5 - 1.3 = -0.8$, $h_6 = 1$

Case A: 왼쪽으로 보낼 경우

- $G_L = -3.05$, $H_L = 4$
- $G_R = -1.00$, $H_R = 1$

$$\text{Gain}_{6,\text{left}} = \frac{1}{2}\left(\frac{(-3.05)^2}{4} + \frac{(-1)^2}{1} - \frac{(-4.05)^2}{5}\right) = 0.0229$$

Case B: 오른쪽으로 보낼 경우

- $G_L = -2.25$, $H_L = 3$
- $G_R = -1.8$, $H_R = 2$

$$\text{Gain}_{6,\text{right}} = \frac{1}{2}\left(\frac{(-2.25)^2}{3} + \frac{(-1.8)^2}{2} - \frac{(-4.05)^2}{5}\right) = 0.0135$$

결론: ID 6은 왼쪽이 더 유리함

요약

결측 ID	Y 값	왼쪽 Gain	오른쪽 Gain	더 나은 방향
ID 5	1.4	0.0181	0.0240	오른쪽
ID 6	1.3	0.0229	0.0135	왼쪽

빅데이터와 금융자료분석 프로젝트 (Team 4)

XGboost 알고리즘을 활용한 은행 대출의 부도 여부 예측 모델 구축

강상묵(20259013) 김형환(20249132) 유석호(20249264) 이현준(20249349) 최영서(20249430) 최재필(20249433)

1. 프로젝트 개요

본 프로젝트는 **여러 데이터 전처리 기법(결측치, 이상치, 특성공학 등)과 머신러닝 알고리즘(이상치 분류, 차원축소, XGBoost 등)을 실제 금융데이터에 적용**해보고 시사점을 도출하기 위해 작성되었습니다.

이를 위해 미국 Lending Club의 P2P 대출 데이터를 사용하였으며, 전반적인 워크플로우는 아래와 같습니다.

1. 데이터의 구조, 특성 파악 (EDA)
2. 데이터의 전처리 (특성에 따른 칼럼 가공, 문자형 변수 처리, 결측치 및 이상치 처리, 변수 선택)
3. XGBoost 알고리즘을 이용한 대출 연체여부 예측 모델 구축 및 평가
    - 샘플링, 모델 튜닝, 성과 평가, 변수 중요도 분석(SHAP), Cat/LightGBM 등 다른 모델과 비교

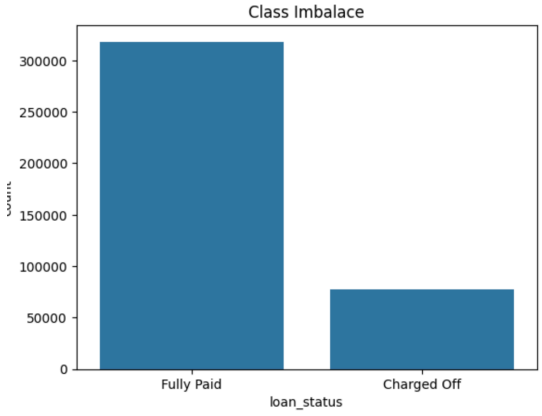
2. 데이터의 구조, 특성 (EDA)

데이터의 수집, 기본구조

미국 소재의 P2P 대출 전문은행인 Lending Club의 '07~'20년 대출 데이터를 사용하였습니다. (출처 : Kaggle)

약 40만개의 데이터로, 목적변수인 대출상태를 포함해 전체 27개의 칼럼(수치형 12 + 문자형 15)으로 이루어져있으며, 목적변수는 정상(상환, Fully paid) 및 부도(연체, Charged off)로 이진분류 문제입니다.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 396030 entries, 0 to 396029
Data columns (total 27 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   loan_amnt             396030 non-null  float64
 1   term                  396030 non-null  object 
 2   int_rate              396030 non-null  float64
 3   installment           396030 non-null  float64
 4   grade                 396030 non-null  object 
 5   sub_grade             396030 non-null  object 
 6   emp_title             373103 non-null  object 
 7   emp_length            377729 non-null  object 
 8   home_ownership        396030 non-null  object 
 9   annual_inc            396030 non-null  float64
 10  verification_status   396030 non-null  object 
 11  issue_d               396030 non-null  object 
 12  loan_status           396030 non-null  object 
 13  purpose               396030 non-null  object 
 14  title                 394274 non-null  object 
 15  dti                   396030 non-null  float64
 16  earliest_cr_line      396030 non-null  object 
 17  open_acc              396030 non-null  float64
 18  pub_rec               396030 non-null  float64
 19  revol_bal             396030 non-null  float64
 20  revol_util            395754 non-null  float64
 21  total_acc             396030 non-null  float64
 22  initial_list_status   396030 non-null  object 
 23  application_type      396030 non-null  object 
 24  mort_acc              358235 non-null  float64
 25  pub_rec_bankruptcies  395495 non-null  float64
 26  address               396030 non-null  object 
dtypes: float64(12), object(15)
memory usage: 81.6+ MB
```



데이터의 특성

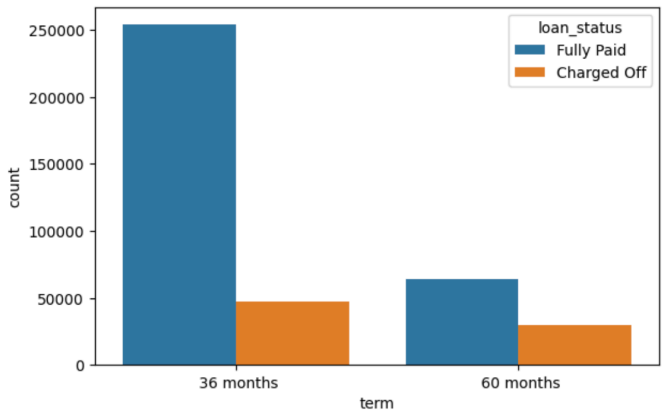
데이터의 각 칼럼별 특징을 알아보고, 적절한 전처리 방법을 탐색해보았습니다.

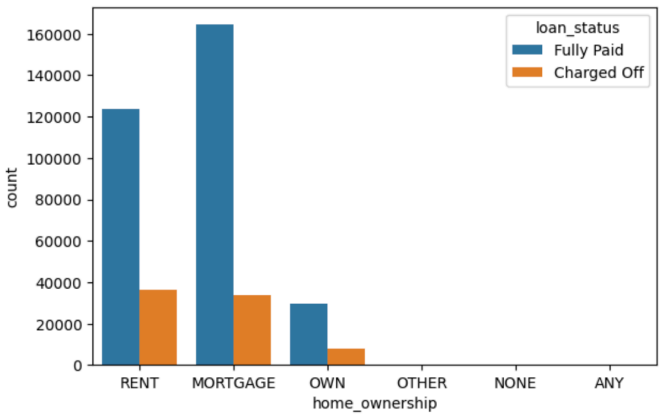
먼저 **수치형 변수**입니다. 결측치 및 이상치 처리는 별도 진행 예정으로 따로 다루지 않겠습니다.

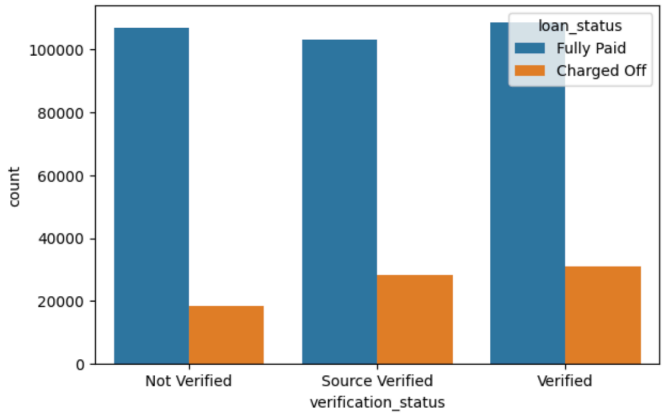
상관관계 행렬을 Heatmap으로 살펴보았습니다. 대체적으로 변수들 간 **상관관계가 미미**하였으며, **일부 상관계수가 높은 변수**들은 변수선택 과정에서 제외하는 등 **별도의 전처리 과정을 통해 다중공선성 문제를 해결**할 계획입니다.

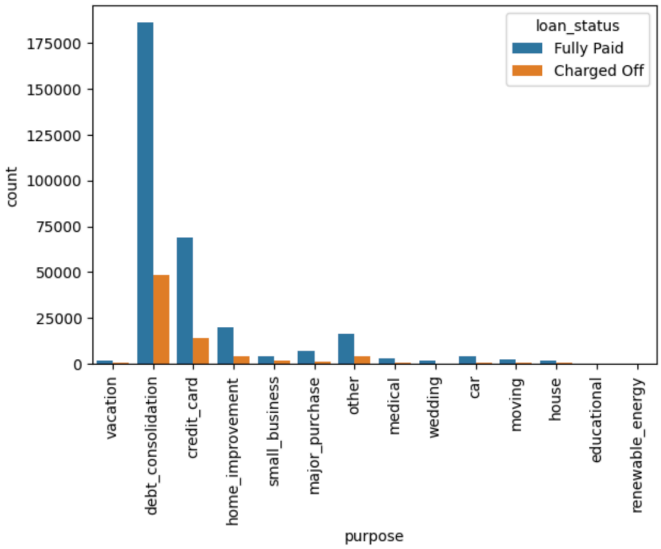
다음으로 **문자형 변수**입니다. 목적변수와 관련이 있는 것으로 보이는 주요 예시만 살펴보겠습니다.

먼저, **대출기간(term), 집보유형태(home_ownership), 대출목적(purpose)이 영향**을 미치는 것으로 추정됩니다.

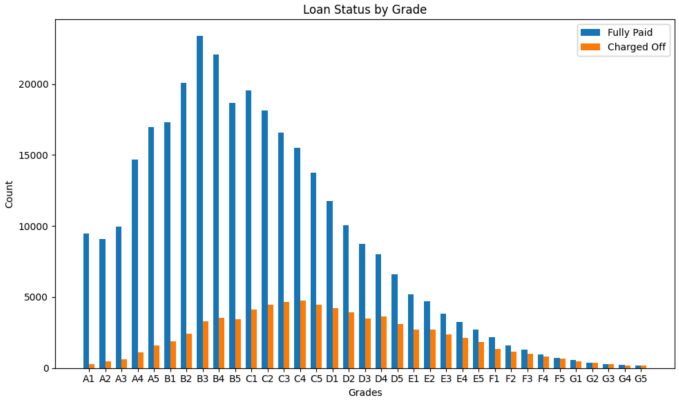








다음으로, **신용등급(A1~G5)에 따라 부도율이 높아지는 추이**를 보였으며, 문자형 변수들 중 **일부는 고유값이 너무 많아 분석에서 제외**하는 것이 효과적일 것으로 보입니다.




```
term                   2
grade                  7
sub_grade             35
emp_title         173105
emp_length            11
home_ownership         6
verification_status    3
issue_d              115
loan_status            2
purpose               14
title              48816
earliest_cr_line     684
initial_list_status    2
application_type       3
address           393700
dtype: int64
```

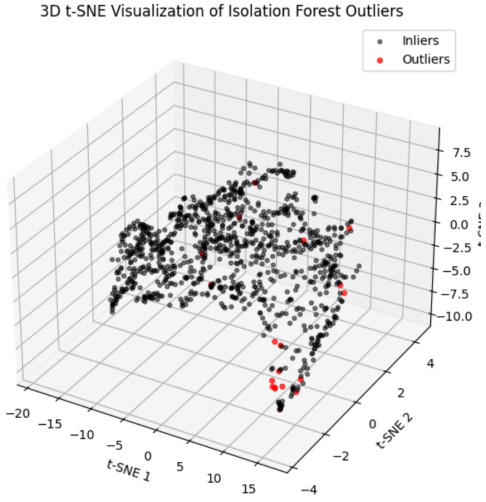
3. 데이터의 전처리

데이터의 전처리는 아래의 과정으로 실시하였습니다.

1. 분석에 적합하도록 칼럼 변환 및 통합, 제거
    - 변환/통합 : 주소(address)는 우편번호(zip_code)만 추출하고 제거, 대출기간(term, 36month 등)은 수치형으로 변환, 집 소유여부(home_ownership)의 극소수값들은 Other로 통합
    - 불필요한 noise 방지를 위해 100개 이상의 고유값을 가진 칼럼 제거 : 직업(title), 직업글자수(emp_title), 발행일(issue_d), 최초연도(earliest_cr_line)
    - 다른 변수와 중복되거나 추론 가능한 칼럼 제거 : 신용점수-대분류(grade), 근속연수(emp_length)
2. 문자형 변수 처리 : 순서가 있거나 이진변수인 경우 라벨인코딩, 단순 점주인 경우 원핫인코딩 적용
    - 라벨인코딩 : 목적변수(이진), 신용점수(순서 존재) / 원핫인코딩 : 이외의 문자형 변수
3. 수치형 변수의 결측치 및 이상치 처리 : 중간값 처리 및 1% 이상치 제거
    - 결측치 : 변수간 상관관계가 미미하고, 이후 Boruta를 적용 예정이므로 예측형 모델보다는 중간값을 채택
    - 이상치 : 고차원, 많은 샘플(약 40만)을 고려, 분포에 대한 가정이 불필요한 Isolation forest 기법 채택
4. 변수 선택을 통해 분석에 적합한 최종 데이터 가공 : Boruta 알고리즘 적용
    - 일부 변수간 상관관계가 존재하는 점을 고려, 최적의 변수 조합을 찾고자 Boruta 알고리즘 채택
    - 원핫인코딩 대상 변수를 제외한 13개(수치형+라벨)에 알고리즘을 적용한 결과 11개의 변수를 선택하였고, 원핫인코딩 대상 변수와 결합하여 최종 데이터 구성

i Isolation Forest 검증(T-SNE 적용) 및 최종 데이터 구성

수치형 변수에 **T-SNE를 적용하여 3차원으로 축소**한 결과, **이상치 제거(Isolation Forest)가 적절히 작동**하였으며, **Boruta 알고리즘으로 변수 선택**까지 마친 후 **최종 데이터**는 **7개의 문자형 변수**(원핫인코딩 6 + 라벨인코딩 1) 및 **9개의 수치형 변수, 1개의 목적변수(이진분류)**로 구성되어 있습니다.




```
<class 'pandas.core.frame.DataFrame'>
Index: 274448 entries, 3412 to 121958
Data columns (total 41 columns):
 #   Column                              Non-Null Count   Dtype
---  ------                              --------------   -----
 0   loan_amnt                           274448 non-null  float64
 1   term                                274448 non-null  int64
 2   int_rate                            274448 non-null  float64
 3   installment                         274448 non-null  float64
 4   sub_grade                           274448 non-null  int64
 5   annual_inc                          274448 non-null  float64
 6   dti                                 274448 non-null  float64
 7   revol_bal                           274448 non-null  float64
 8   revol_util                          274448 non-null  float64
 9   total_acc                           274448 non-null  float64
 10  mort_acc                            274448 non-null  float64
 11  home_ownership_OTHER                274448 non-null  bool
 12  home_ownership_OWN                  274448 non-null  bool
 13  home_ownership_RENT                 274448 non-null  bool
 14  verification_status_Source Verified 274448 non-null  bool
 15  verification_status_Verified        274448 non-null  bool
 16  purpose_credit_card                 274448 non-null  bool
 17  purpose_debt_consolidation          274448 non-null  bool
 18  purpose_educational                 274448 non-null  bool
 19  purpose_home_improvement            274448 non-null  bool
 20  purpose_house                       274448 non-null  bool
 21  purpose_major_purchase              274448 non-null  bool
 22  purpose_medical                     274448 non-null  bool
 23  purpose_moving                      274448 non-null  bool
 24  purpose_other                       274448 non-null  bool
 25  purpose_renewable_energy            274448 non-null  bool
 26  purpose_small_business              274448 non-null  bool
 27  purpose_vacation                    274448 non-null  bool
 28  purpose_wedding                     274448 non-null  bool
 29  initial_list_status_w               274448 non-null  bool
 30  application_type_INDIVIDUAL         274448 non-null  bool
 31  application_type_JOINT              274448 non-null  bool
 32  zip_code_05113                      274448 non-null  bool
 33  zip_code_11650                      274448 non-null  bool
 34  zip_code_22690                      274448 non-null  bool
 35  zip_code_29597                      274448 non-null  bool
 36  zip_code_30723                      274448 non-null  bool
 37  zip_code_48052                      274448 non-null  bool
 38  zip_code_70466                      274448 non-null  bool
 39  zip_code_86630                      274448 non-null  bool
 40  zip_code_93700                      274448 non-null  bool
dtypes: bool(30), float64(9), int64(2)
memory usage: 33.0 MB
```

4. 대출 연체여부 예측 모델 구축 및 평가

ADASYN을 이용한 오버샘플링

모델링에 앞서, 클래스 불균형 문제는 **ADASYN을 통한 오버샘플링**으로 해결하였습니다. 과대평가, 과적합을 방지하고 **검증 무결성을 위해 CV 과정의 훈련 데이터에만 오버샘플링**하였으며, (`imblearn.pipeline` 활용) 이를 통해 **검증은 항상 원본데이터로만 진행**됩니다.

XGBoost 모델 구축

앞서 구성한 40개 변수로 **"대출 연체 여부"를 예측하는 모델을 XGBoost 알고리즘**을 통해 구축하였으며, **모델 튜닝은 2단계 최적화 접근법을 적용**하였습니다. 이러한 방식은 **과적합을 방지하고 정해진 계산자원 하에서 최대한 공정하게 파라미터를 비교**할 수 있는 장점이 있습니다.

1단계: 초기 하이퍼파라미터 탐색 - 상대적으로 높은 학습률(0.1)과 고정된 n_estimators 값으로 하이퍼파라미터 조합을 탐색 - 각 조합이 동일한 학습 기회(같은 트리 개수)를 갖도록 보장 (CV 내에서 ADASYN 오버샘플링 적용)

2단계: 최적 모델 미세 조정 - 1단계에서 찾은 최적 하이퍼파라미터에 낮은 학습률(0.01)과 높은 n_estimators(10000) 적용 - 조기종료를 적용(50)하여 최적의 트리 개수 결정하고, 전체 훈련/검증 데이터를 사용하여 모델 학습

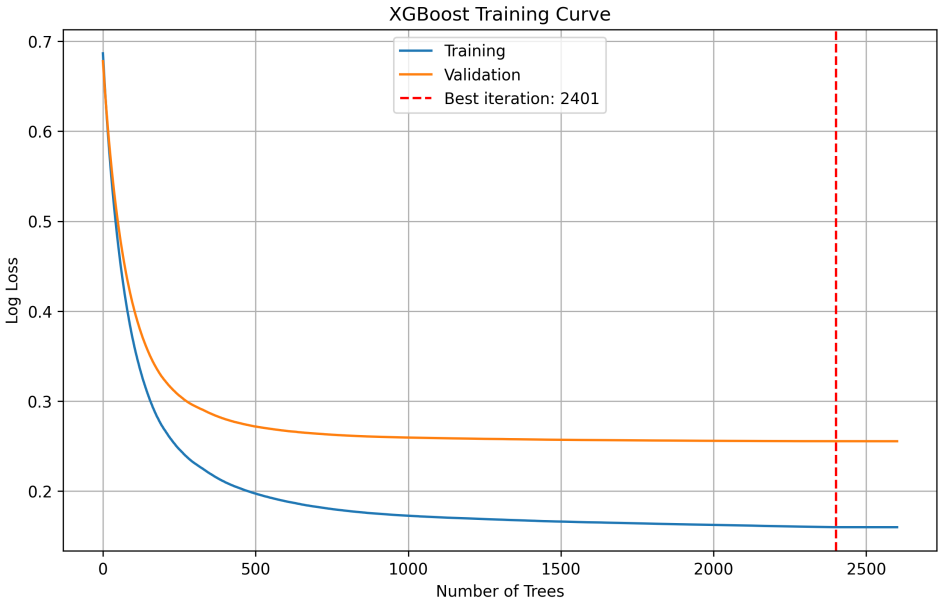


Figure 2.1: 2단계 접근법의 최종 모델 학습곡선

모델 일반화성능 평가

클래스 불균형 문제를 고려하여 최종 모델의 **일반화 성능을 F1-Score를 중심**으로 평가하도록 하겠습니다. **F1-score는 약 0.94, ROC-AUC는 약 0.91로 실제 연체 여부를 잘 예측**하는 것으로 나타났습니다.

특히, 모델 튜닝 과정에서 1단계 하이퍼파라미터 탐색시 최고 F1-Score가 0.93544였는데, 일반화 성능은 그와 동일한 수준이므로 **과적합 방지를 위한 2단계 최적화 기법이 효과적**인 것을 알 수 있습니다.

```
<< 최종 XGBoost 모델 성능 >>
F1 Score: 0.9354695016927006
ROC AUC: 0.9081721636844388

분류 보고서:
              precision    recall  f1-score   support

           0       0.92      0.48      0.63     23302
           1       0.89      0.99      0.94     95507

    accuracy                           0.89    118809
   macro avg       0.90      0.74      0.78    118809
weighted avg       0.89      0.89      0.88    118809
```

여러 Gradiant boosting 계열의 알고리즘과 비교

주로 사용한 XGBoost 이외에도 다양한 Gradiant boosting 계열 및 앙상블 알고리즘이 존재합니다.

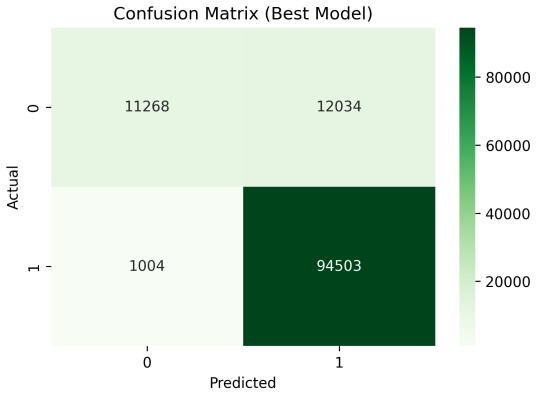
- **XGBoost**: Regularization과 트리 구조 최적화에 강점을 가진 Gradient Boosting 모델 |
- **CatBoost**: 범주형 변수 자동 인식 기능이 있는 Gradient Boosting 기반 모델
- **LightGBM**: 빠른 학습 속도와 낮은 메모리 사용의 Gradient Boosting 기반 모델
- **Soft Voting**: CatBoost, LightGBM의 예측 확률 평균을 통한 결합(앙상블) 모델
- **Stacking**: CatBoost, LightGBM의 예측 결과를 Logistic Regression에 전달하는 메타 모델 기반 앙상블

XGBoost와 유사한 방식으로 각 모델을 튜닝, 훈련하였으며 **일반화성능은 유사한 수준**이였습니다. F1-Score는 앙상블(Soft Voting) 모델이, ROC-AUC 점수는 XGBoost가 가장 우수하였습니다.

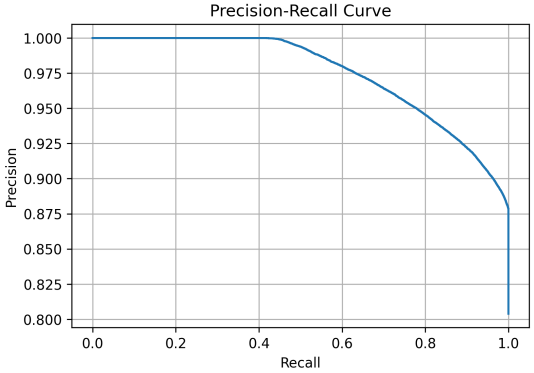
모델	F1 Score	ROC AUC	모델	F1 Score	ROC AUC
CatBoost	0.9355	0.9073	**Soft Voting**	0.9356	0.9072
LightGBM	0.9353	0.9066	**Stacking**	0.9330	0.9072

그러나, **샘플이 적은 "부도"인 경우, 예측 성능이 다소 떨어지는 모습**이 관측되었습니다.

부도의 절반 이상이 정상으로 분류되었으며 모든 모델에 동일한 문제가 있는 것으로 볼 때, **데이터의 한계**인 것으로 보입니다. 또는 **신경망 계열을 적용해보는 것도 개선방법**이 될 수 있습니다.



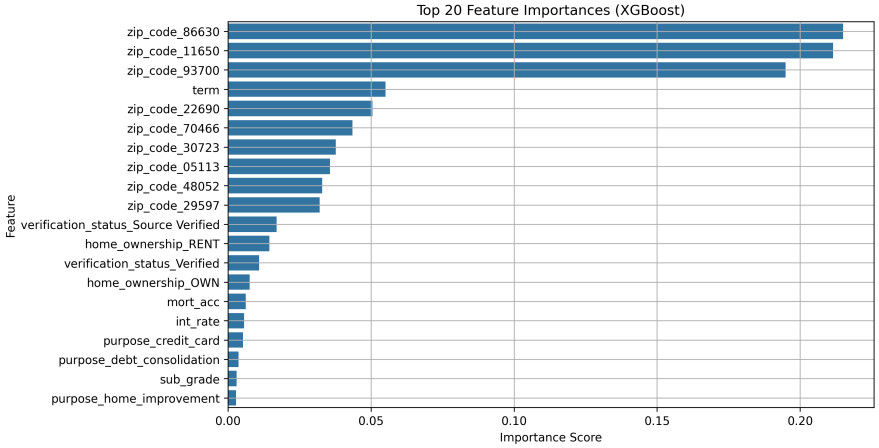
(a) XGBoost Confusion Matrix



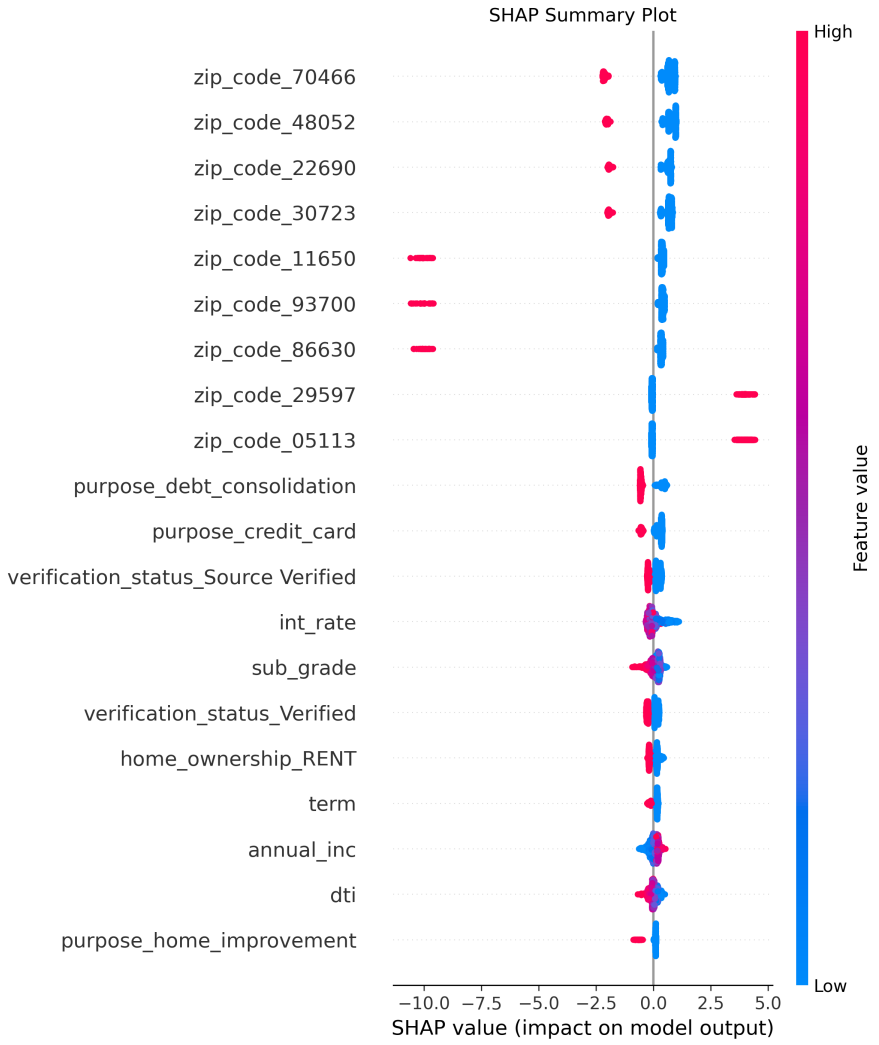
(a) XGBoost PRCurve

변수 중요도(Feature Importance) 분석

기본 변수 중요도 산출 결과 부도 여부에는 예상 외로 **주거지가 큰 영향**을 미치는 것을 확인할 수 있었으며, 이외에도 **대출기간 등이 분류에 영향**을 미치는 것을 알 수 있었습니다.



보다 **상세한 변수 중요도 분석을 위해 SHAP(SHapley Additive exPlanations) 분석**을 실시하여 **각 변수의 중요도, 예측에 미치는 영향(방향성, 정도)을 분석**하였습니다.



전체적인 중요도는 **주거지가 큰 영향을 미친다는 점에서 유사**하였으며, 변수별 특징은 아래와 같습니다.

- 상위 7개는 해당 지역에 주거(1) 시, 연체(0)가 많으므로 집값/소득수준이 낮은 주거지로 추정
- 하위 2개는 집값/소득수준이 높은 주거지로 추정
- 대출목적이 부채상환/신용카드, 대출이자가 높고, 대출기간이 길고, 소득이 낮을수록 연체 예측 가능성 증가
- 이 외에도 대부분의 변수가 대출과 관련된 일반적인 직관과 동일한 것을 알 수 있었음

시사점

이번 프로젝트를 통해 실제 은행의 데이터를 살펴보고, 대출의 부도 확률 예측 모델을 구축해보았습니다.

먼저 **실제 데이터를 전처리**하는 과정에서 발생하는 결측치, 이상치, 적합하지 않은 변수 분류 등의 **문제점을 실제로 경험**할 수 있었고 Isolation Forest 및 T-SNE, Boruta 알고리즘을 적용해보면서 **각 알고리즘이 어떻게 작동하는지, 어떤 방식으로 문제를 해결하고 활용**되는지 알 수 있었습니다.

또한, **XGBoost 알고리즘**이 **금융데이터 예측에 강력한 성능**을 가진 것을 확인하였고, 모델 성능에는 알고리즘 선택뿐만아니라 **하이퍼파라미터 튜닝 방법(2단계 최적화) 클래스 불균형 해소(oversampling), 변수 선별(boruta) 등이 매우 중요**하다는 것을 느꼈습니다.

결과적으로 은행 대출의 **연체여부에는 주거지가 매우 큰 영향**을 미친것으로 나타났으며, 이는 **주거지에 집값/주거형태/소득/신용점수/직업 등 종합적인 요소가 모두 반영**되어있기 때문인 것으로 추정됩니다. 이외에도 **이자율, 대출목적/기간 등이 연체 여부 예측에 영향**을 미치는 것으로 나타났으며, **그 영향은 일반적인 직관과 동일**하였습니다.

데이터 자체의 한계점으로 인해 소수 표본에 대한 학습이 부족하여 예측력이 다소 떨어지는 한계점이 있었으나, **전반적으로 수업시간에 다룬 여러 알고리즘을 통해 이론이 실제 세상에 적용되는 과정을 이해**할 수 있었습니다. 또한, **분석에 적합한 데이터를 구하고 전처리하는 것이 매우 중요**하다는 것을 알게 된 프로젝트였습니다.

Part II

알고리즘 거래전략('25 봄)

알고리즘 거래전략과 고빈도 금융

3 알고리즘 거래전략 기말과제

20249132 김형환

```
from tqdm import tqdm
from statsmodels.tsa.stattools import adfuller
from itertools import combinations, product
from joblib import Parallel, delayed
import statsmodels.api as sm
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
# 기존 데이터를 R로 가공하여 2022~2024의 snp500구성종목 데이터만 남기고, long form으로 변환
snp500_return = pd.read_csv("data/snp500_return.csv")
snp500_return['date'] = pd.to_datetime(snp500_return['date'])
snp500_return.head()
```

	date	permno	ticker	comnam	return
0	2022-01-03	10104	ORCL	ORACLE CORP	0.007912
1	2022-01-03	10107	MSFT	MICROSOFT CORP	-0.004668
2	2022-01-03	10138	TROW	T ROWE PRICE GROUP INC	-0.010476
3	2022-01-03	10145	HON	HONEYWELL INTERNATIONAL INC	-0.008201
4	2022-01-03	10516	ADM	ARCHER DANIELS MIDLAND CO	0.004439

```
# 공적분 관계 검정을 위해 wide form 생성, 추후 전략 성능평가를 위해 24년 1년간은 테스트 목적으로 분할
snp500_return_train = snp500_return[snp500_return["date"] < "2024-01-01"]
snp500_return_test  = snp500_return[snp500_return["date"] >= "2024-01-01"]

snp500_return_wide = snp500_return_train.pivot(index='date', columns='permno', values='return'
snp500_return_wide_test = snp500_return_test.pivot(index='date', columns='permno', values='ret

snp500_prices_wide = (1 + snp500_return_wide).cumprod()
snp500_prices_wide_test = (1 + snp500_return_wide_test).cumprod()
```

```
print(snp500_return_wide.info())
print(snp500_return_wide_test.info())
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 501 entries, 2022-01-03 to 2023-12-29
Columns: 456 entries, 10104 to 93436
dtypes: float64(456)
memory usage: 1.7 MB
None
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 252 entries, 2024-01-02 to 2024-12-31
Columns: 456 entries, 10104 to 93436
dtypes: float64(456)
memory usage: 899.7 KB
None
```

```
# 공적분 검정 이전 계산량 효율화를 위한 사전 스크리닝
# 1. 단일 ADF검정 활용
alpha = 0.05

tqdm.pandas(desc="ADF Test")
pvals = snp500_prices_wide.progress_apply(lambda ts: adfuller(ts)[1])
non_sta = pvals[pvals >= alpha].index.tolist()
filtered_prices_wide = snp500_prices_wide[non_sta]

print(f"Number of total stocks : {len(pvals)}")
print(f"Number of non-stationary stocks(p<{alpha}) : {len(non_sta)}")
```

```
ADF Test: 100%|      | 456/456 [00:03<00:00, 130.26it/s]

Number of total stocks : 456
Number of non-stationary stocks(p<0.05) : 370
```

```
# 공적분 검정 이전 계산량 효율화를 위한 사전 스크리닝
# 2. 상관계수 활용
corr_threshold = 0.3

corr = filtered_prices_wide.corr().abs()
pairs = [
```

```
    (i, j)
    for i, j in combinations(filtered_prices_wide.columns, 2)
    if corr.loc[i, j] > corr_threshold]

print(f"Number of whole pairs : {len(list(combinations(filtered_prices_wide.columns, 2)))}")
print(f"Number of correlated(>{corr_threshold}) pairs : {len(pairs)}")
```

```
Number of whole pairs : 68265
Number of correlated(>0.3) pairs : 43997
```

```
# Engle-Granger 테스트를 통해 공적분 관계 검정정

alpha_coint = 0.01
min_obs = 50

def test_pair(pair):
    t1, t2 = pair
    df = filtered_prices_wide[[t1, t2]].dropna()
    if len(df) < min_obs:
        return None
    y = df[t1]
    x = sm.add_constant(df[t2])
    model = sm.OLS(y, x).fit()
    resid = model.resid
    try:
        pval = adfuller(resid, autolag=None)[1]
        if pval < alpha_coint:
            return {"permno_1": t1, "permno_2": t2, "pvalue": pval}
    except:
        return None
    return None

results = Parallel(n_jobs=-1, verbose=5)(
    delayed(test_pair)(pair) for pair in pairs
)

results = [r for r in results if r is not None]
coint_pairs = pd.DataFrame(results)
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.
[Parallel(n_jobs=-1)]: Done  40 tasks      | elapsed:    4.4s
[Parallel(n_jobs=-1)]: Done 168 tasks      | elapsed:    4.7s
[Parallel(n_jobs=-1)]: Done 2080 tasks      | elapsed:    6.0s
[Parallel(n_jobs=-1)]: Done 7264 tasks      | elapsed:    9.3s
[Parallel(n_jobs=-1)]: Done 13600 tasks      | elapsed:   13.6s
[Parallel(n_jobs=-1)]: Done 21088 tasks      | elapsed:   17.4s
[Parallel(n_jobs=-1)]: Done 29728 tasks      | elapsed:   22.1s
[Parallel(n_jobs=-1)]: Done 39520 tasks      | elapsed:   27.4s
[Parallel(n_jobs=-1)]: Done 43966 out of 43997 | elapsed:   30.4s remaining:    0.0s
[Parallel(n_jobs=-1)]: Done 43997 out of 43997 | elapsed:   30.5s finished
```

```
print(f"Number of cointegrated(significant level={alpha_coint:.0%}) pairs : {len(coint_pairs)}
```

```
Number of cointegrated(significant level=1%) pairs : 2565
```

```
# 상위 5개 공적분 페어 추출
comnam_map = snp500_return.drop_duplicates("permno").set_index("permno")["comnam"].to_dict()
top5_pairs = coint_pairs.sort_values("pvalue").head(5).copy()
top5_pairs["comnam_1"], top5_pairs["comnam_2"] = top5_pairs["permno_1"].map(comnam_map), top5_

top5_pairs
```

	permno_1	permno_2	pvalue	comnam_1	comnam_2
320	11850	93246	9.981788e-07	EXXON MOBIL CORP	GENERAC HOLDINGS INC
2308	76605	77730	1.170548e-05	AUTOZONE INC	TYSON FOODS INC
376	13407	86580	1.595093e-05	META PLATFORMS INC	NVIDIA CORP
1419	24643	52708	1.624042e-05	HOWMET AEROSPACE INC	LENNAR CORP
2495	87137	90215	2.325270e-05	DEVON ENERGY CORP NEW	SALESFORCE INC

```
# Pair들에 대한 반감기, Historical Mean Crossing, 헤지비율, Hurst 지수, Pearson Corr 계산
pair_list   = list(coint_pairs[["permno_1","permno_2","pvalue"]].itertuples(index=False))
total_pairs = len(pair_list)

records = []
for permno_1, permno_2, pval in tqdm(pair_list, total=total_pairs, desc="Compute metrics"):

    pair = pd.concat([filtered_prices_wide[permno_1], filtered_prices_wide[permno_2]], axis=1,
    if len(pair) < 252:
```

```
        continue2
    y1, y2 = pair.iloc[:,0], pair.iloc[:,1]

    # OLS로 hedge ratio  추정
    X    = sm.add_constant(y2.values)
    res = sm.OLS(y1.values, X).fit()
    beta = res.params[1]

    spread = y1 - beta * y2

    # Hurst exponent
    lags = np.array([2,5,10,20,50])
    tau  = np.array([np.std(spread.diff(l).dropna()) for l in lags])
    H    = np.polyfit(np.log(lags), np.log(tau), 1)[0]

    # Half-life
    spread_lag    = spread.shift(1).dropna()
    spread_ret    = spread.diff().dropna().loc[spread_lag.index]
    reg           = sm.OLS(spread_ret.values, sm.add_constant(spread_lag.values)).fit()
    b_coef        = reg.params[1]
    phi_ar        = b_coef + 1
    half_life     = -np.log(2) / np.log(phi_ar)

    # Historical mean crossings per year
    m               = spread.mean()
    centered        = spread - m
    signs           = np.sign(centered)
    crossings       = np.sum(signs.diff().fillna(0) != 0)
    years           = (spread.index[-1] - spread.index[0]).days / 365.25
    mean_cross_year = crossings / years

    # Pearson Corr
    ret_pair = snp500_return_wide.loc[:, [permno_1, permno_2]].dropna(how="any")
    pearson_corr = ret_pair.corr().iloc[0,1] if len(ret_pair) > 1 else np.nan

    records.append({
        "permno_1":        permno_1,
        "permno_2":        permno_2,
        "pvalue":          pval,
```

```
        "hedge_ratio":     beta,
        "hurst":           H,
        "half_life":       half_life,
        "mean_cross_year": mean_cross_year,
        "pearson_corr":    pearson_corr
    })

metrics_df_1pct = pd.DataFrame(records)
metrics_df_1pct["comnam_1"] = metrics_df_1pct["permno_1"].map(comnam_map)
metrics_df_1pct["comnam_2"] = metrics_df_1pct["permno_2"].map(comnam_map)
```

```
Compute metrics: 100%|      | 2565/2565 [00:23<00:00, 111.44it/s]
```

```
# Hurst,Half-life,Mean-crossings,pearson_corr 필터링
filtered_pairs = metrics_df_1pct[
    (metrics_df_1pct["hurst"]           < 0.5)   &
    (metrics_df_1pct["half_life"]       > 1)     &
    (metrics_df_1pct["half_life"]       < 30)    &
    (metrics_df_1pct["mean_cross_year"] > 5)    &
    (metrics_df_1pct["pearson_corr"] > 0.5)
].copy()

# 점수 계산 및 최우수 페어 추출
filtered_pairs["score"] = (
    -filtered_pairs["pvalue"].rank() +
    -filtered_pairs["half_life"].rank() +
    filtered_pairs["mean_cross_year"].rank() +
    filtered_pairs["pearson_corr"].rank()
)

best5_pairs = filtered_pairs.sort_values("score", ascending=False).iloc[0:5]

print(f"Number of Filtered pairs : {len(filtered_pairs)}")
best5_pairs
```

```
Number of Filtered pairs : 322
```

	permno_1	permno_2	pvalue	hedge_ratio	hurst	half_life	mean_cross_year	pearson_corr
853	19286	78987	0.000158	0.680322	0.314507	7.191855	32.242759	0.605119
340	12084	48486	0.001111	0.596882	0.360651	10.127506	29.220000	0.792955
2107	58819	64936	0.000495	0.300429	0.402952	12.482854	29.220000	0.761218
1178	23931	24109	0.000785	0.848911	0.393015	12.980947	29.220000	0.886921
848	19286	60871	0.001356	0.737086	0.314549	8.868845	29.723793	0.621187

```
# 최적 쌍을 이용하여 전략 구성. 성과가 가장 좋은 진입청산기준을 선정해서 test데이터로 평가 실시
pair = best5_pairs.iloc[0]
p1, p2 = pair.permno_1, pair.permno_2
hedge_ratio = pair.hedge_ratio
comnam1, comnam2 = pair.comnam_1, pair.comnam_2

# 가격 데이터
price_is = snp500_prices_wide[[p1, p2]].dropna()
price_is.columns = [comnam1, comnam2]

# 파라미터 후보
entry_z_list = [0.5 ,1.0, 1.5, 2.0, 3.0]
exit_z_list = [0.0, 0.1, 0.3, 0.5]
transaction_cost = 0.0003  # 거래비용 3bp

results = []

for entry_z, exit_z in product(entry_z_list, exit_z_list):
    spread = price_is[comnam1] - hedge_ratio * price_is[comnam2]
    mu, sigma = spread.mean(), spread.std()
    z = (spread - mu) / sigma

    long_entry  = z < -entry_z
    short_entry = z >  entry_z
    exit_signal = z.abs() < exit_z

    # 포지션 설정
    pos = np.where(long_entry, 1,
          np.where(short_entry, -1,
          np.where(exit_signal, 0, np.nan)))
    pos = pd.Series(pos, index=z.index).ffill().fillna(0)
```

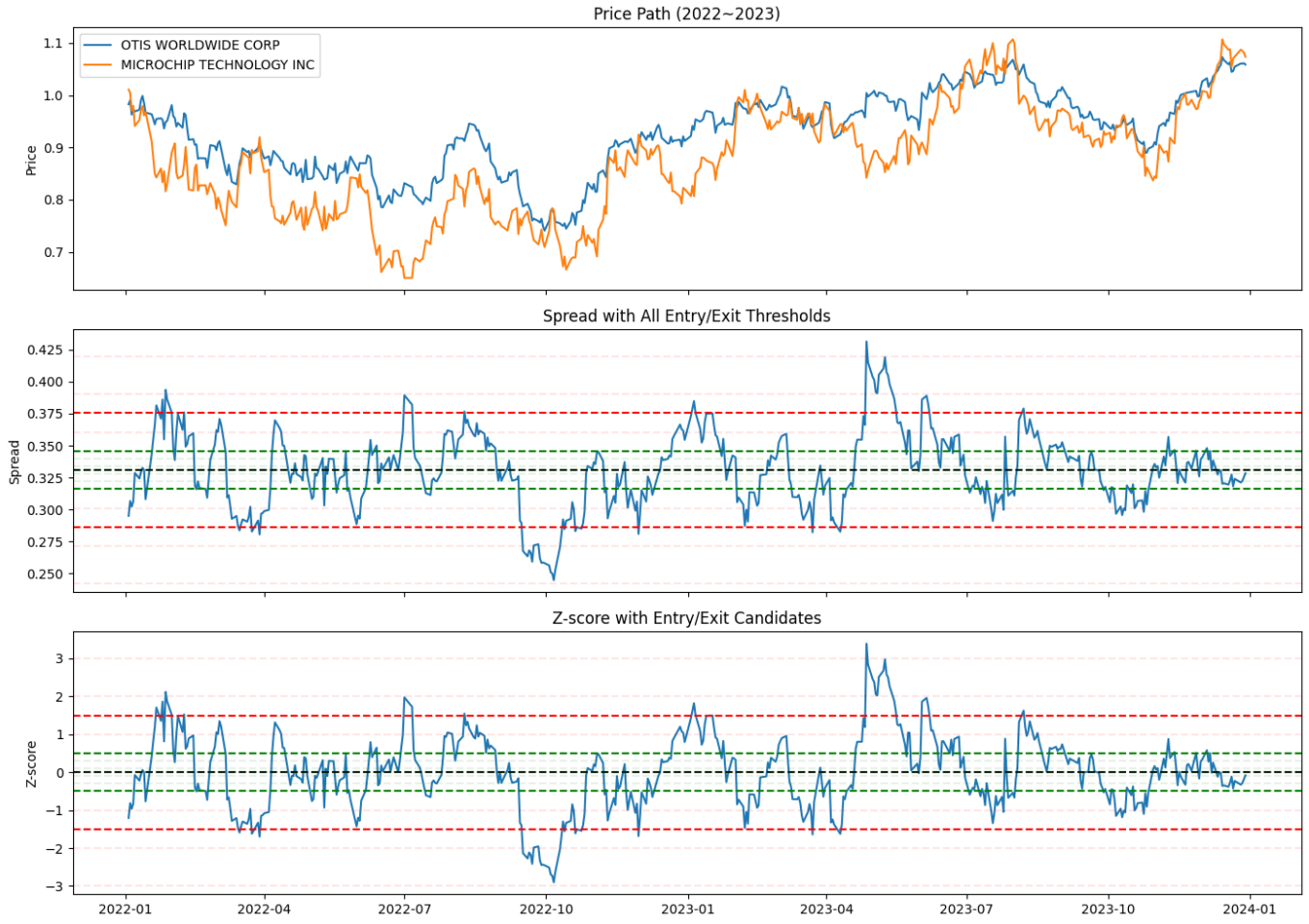
```
    # 수익률 계산
    ret = price_is.pct_change().dropna()
    strat_ret = pos.shift() * (ret[comnam1] - hedge_ratio * ret[comnam2])

    # 거래비용 반영 (포지션 변경 시 비용 발생)
    pos_chg = pos.diff().abs()
    cost = pos_chg.shift().fillna(0) * transaction_cost
    strat_ret_net = strat_ret - cost

    cum_ret = (1 + strat_ret_net).cumprod()
    sharpe = strat_ret_net.mean() / strat_ret_net.std() * np.sqrt(252)
    mdd = (cum_ret / cum_ret.cummax() - 1).min()

    results.append({
        "entry_z": entry_z,
        "exit_z": exit_z,
        "final_return": cum_ret.iloc[-1],
        "sharpe": sharpe,
        "mdd": mdd
    })

# 결과 정리
opt_results = pd.DataFrame(results).sort_values(by="sharpe", ascending=False).reset_index(drop
display(opt_results)
```

	entry_z	exit_z	final_return	sharpe	mdd
0	1.5	0.5	2.015736	3.152895	-0.044008
1	1.5	0.3	2.098776	3.035545	-0.056847
2	0.5	0.5	2.709583	2.929092	-0.088697
3	1.0	0.3	2.527457	2.925949	-0.070058
4	1.5	0.1	2.165035	2.904456	-0.068544
5	1.0	0.5	2.366084	2.890452	-0.070058
6	0.5	0.3	2.716753	2.780591	-0.088697
7	0.5	0.1	2.743286	2.626813	-0.088697
8	1.0	0.0	2.926148	2.523193	-0.093610
9	1.0	0.1	2.314327	2.489392	-0.076772
10	0.5	0.0	2.780223	2.405471	-0.093610
11	1.5	0.0	2.411925	2.112670	-0.102202

	entry_z	exit_z	final_return	sharpe	mdd
12	2.0	0.5	1.262682	1.875213	-0.038795
13	2.0	0.3	1.278799	1.811135	-0.038795
14	3.0	0.5	1.099983	1.519723	-0.031609
15	3.0	0.1	1.114023	1.294801	-0.031609
16	3.0	0.3	1.114023	1.294801	-0.031609
17	2.0	0.1	1.288337	1.231710	-0.101385
18	2.0	0.0	1.496786	1.036650	-0.139779
19	3.0	0.0	1.121741	0.578127	-0.080870

```
# 최적 entry_z, exit_z
entry_z_opt = opt_results.iloc[0]["entry_z"]
exit_z_opt  = opt_results.iloc[0]["exit_z"]

entry_z_list = [0.5 ,1.0, 1.5, 2.0, 3.0]
exit_z_list = [0.0, 0.1, 0.3, 0.5]

spread = price_is[comnam1] - hedge_ratio * price_is[comnam2]
mu, sigma = spread.mean(), spread.std()
z = (spread - mu) / sigma

fig, axs = plt.subplots(3, 1, figsize=(14, 10), sharex=True)

# (1) 가격 경로
axs[0].plot(price_is[comnam1], label=comnam1)
axs[0].plot(price_is[comnam2], label=comnam2)
axs[0].set_ylabel("Price")
axs[0].legend()
axs[0].set_title("Price Path (2022~2023)")

# (2) 스프레드 + 기준선
axs[1].plot(spread, label="Spread")
axs[1].axhline(mu, color="black", linestyle="--", label="Mean")

for ez in entry_z_list:
    alpha = 1.0 if ez == entry_z_opt else 0.1
    axs[1].axhline(mu + sigma * ez, color="red", linestyle="--", alpha=alpha, label=f"+{ez}" i
    axs[1].axhline(mu - sigma * ez, color="red", linestyle="--", alpha=alpha, label=f"-{ez}" i
```

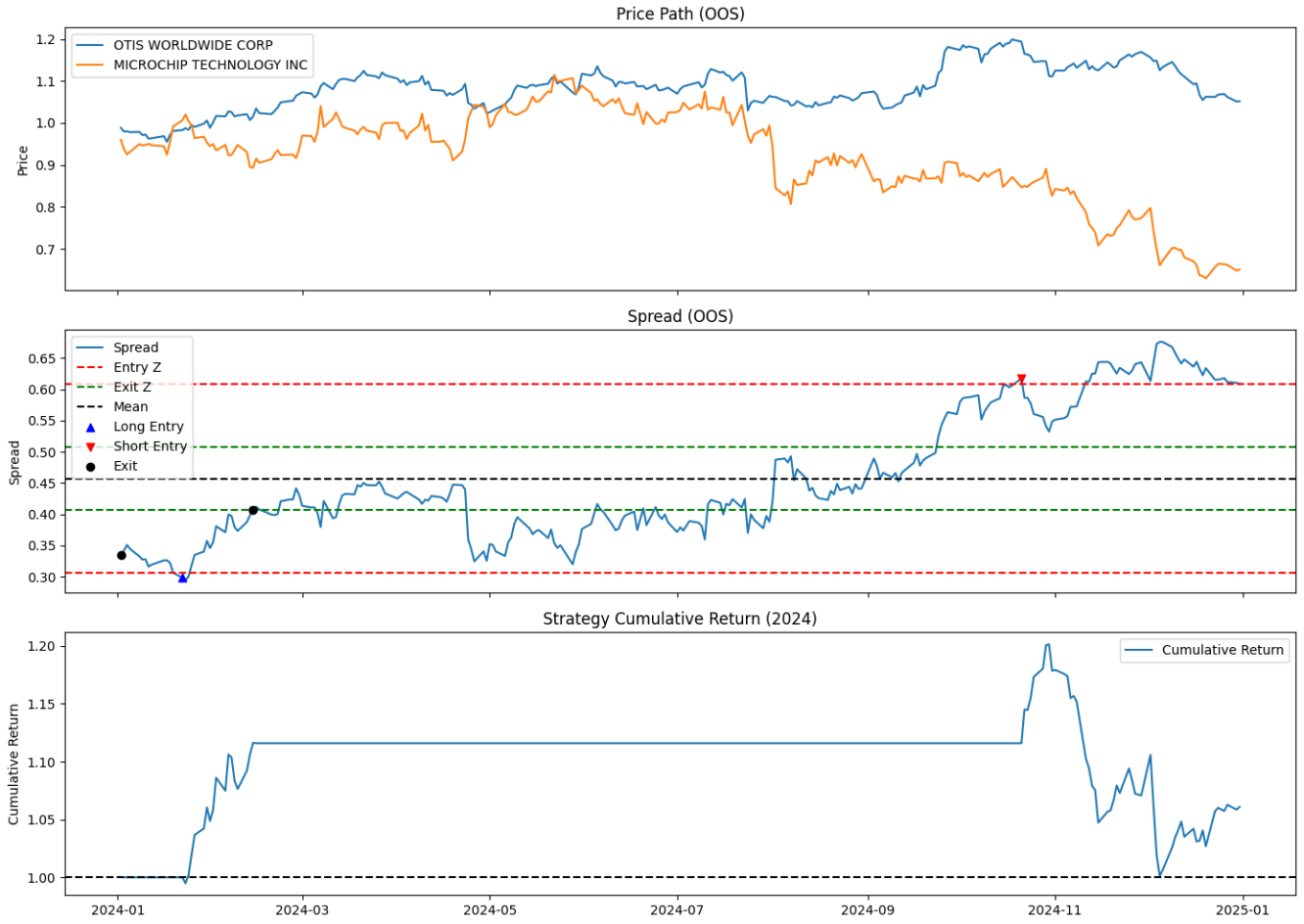
```
for ez in exit_z_list:
    alpha = 1.0 if ez == exit_z_opt else 0.1
    axs[1].axhline(mu + sigma * ez, color="green", linestyle="--", alpha=alpha, label=f"+{ez}"
    axs[1].axhline(mu - sigma * ez, color="green", linestyle="--", alpha=alpha, label=f"-{ez}"

axs[1].set_ylabel("Spread")
axs[1].set_title("Spread with All Entry/Exit Thresholds")

# (3) z-score + 기준선
axs[2].plot(z, label="Z-score")
axs[2].axhline(0, color="black", linestyle="--")

for ez in entry_z_list:
    alpha = 1.0 if ez == entry_z_opt else 0.1
    axs[2].axhline(ez, color="red", linestyle="--", alpha=alpha)
    axs[2].axhline(-ez, color="red", linestyle="--", alpha=alpha)

for ez in exit_z_list:
    alpha = 1.0 if ez == exit_z_opt else 0.1
    axs[2].axhline(ez, color="green", linestyle="--", alpha=alpha)
    axs[2].axhline(-ez, color="green", linestyle="--", alpha=alpha)

axs[2].set_ylabel("Z-score")
axs[2].set_title("Z-score with Entry/Exit Candidates")

plt.tight_layout()
plt.show()
```




```
# OOS 가격 데이터
price_oos = snp500_prices_wide_test[[p1, p2]].dropna()
price_oos.columns = [comnam1, comnam2]

# 최적 파라미터
entry_z = opt_results.iloc[0]["entry_z"]
exit_z  = opt_results.iloc[0]["exit_z"]

# 스프레드 및 z-score
spread_oos = price_oos[comnam1] - hedge_ratio * price_oos[comnam2]
mu, sigma = spread_oos.mean(), spread_oos.std()
z_oos = (spread_oos - mu) / sigma

# 포지션
long_entry  = z_oos < -entry_z
short_entry = z_oos >  entry_z
exit_signal = z_oos.abs() < exit_z

pos = np.where(long_entry, 1,
```

```
      np.where(short_entry, -1,
      np.where(exit_signal, 0, np.nan)))
pos = pd.Series(pos, index=z_oos.index).ffill().fillna(0)

# 수익률 계산
ret = price_oos.pct_change().dropna()
strat_ret = pos.shift() * (ret[comnam1] - hedge_ratio * ret[comnam2])

# 거래비용 차감
pos_chg = pos.diff().abs().fillna(0)
cost = pos_chg.shift().fillna(0) * transaction_cost
strat_ret_net = strat_ret - cost

cum_ret = (1 + strat_ret_net).cumprod()

# 성과지표
sharpe = strat_ret_net.mean() / strat_ret_net.std() * np.sqrt(252)
mdd = (cum_ret / cum_ret.cummax() - 1).min()

# 진입/청산 시점
entry_longs  = ((pos.shift(1) == 0) & (pos == 1))
entry_shorts = ((pos.shift(1) == 0) & (pos == -1))
exits        = ((pos.shift(1) != 0) & (pos == 0))

entry_longs_idx = entry_longs[entry_longs].index
entry_shorts_idx = entry_shorts[entry_shorts].index
exits_idx = exits[exits].index

# 시각화
fig, axs = plt.subplots(3, 1, figsize=(14, 10), sharex=True)

# (1) 가격
axs[0].plot(price_oos[comnam1], label=comnam1)
axs[0].plot(price_oos[comnam2], label=comnam2)
axs[0].set_ylabel("Price")
axs[0].legend()
axs[0].set_title("Price Path (OOS)")

# (2) Spread + Entry/Exit 기준선
```

```
axs[1].plot(spread_oos, label="Spread")
axs[1].axhline(mu + sigma * entry_z, color="red", linestyle="--", label="Entry Z")
axs[1].axhline(mu - sigma * entry_z, color="red", linestyle="--")
axs[1].axhline(mu + sigma * exit_z, color="green", linestyle="--", label="Exit Z")
axs[1].axhline(mu - sigma * exit_z, color="green", linestyle="--")
axs[1].axhline(mu, color="black", linestyle="--", label="Mean")

# 진입/청산 화살표
axs[1].scatter(entry_longs_idx, spread_oos.loc[entry_longs_idx], marker='^', color='blue', lab
axs[1].scatter(entry_shorts_idx, spread_oos.loc[entry_shorts_idx], marker='v', color='red', la
axs[1].scatter(exits_idx, spread_oos.loc[exits_idx], marker='o', color='black', label='Exit',

axs[1].set_ylabel("Spread")
axs[1].legend()
axs[1].set_title("Spread (OOS)")

# (3) 누적 수익률
axs[2].plot(cum_ret, label="Cumulative Return")
axs[2].axhline(1, color="black", linestyle="--")
axs[2].set_ylabel("Cumulative Return")
axs[2].legend()
axs[2].set_title("Strategy Cumulative Return (2024)")

plt.tight_layout()
plt.show()
```




```
print("--- Pair tradind Perfomance in 2024 ---")
print(f" 1. Cummulative return : {cum_ret.iloc[-1]:.4f}")
print(f" 2. Sharpe ratio       : {sharpe:.2f}")
print(f" 3. Max Drawdown       : {mdd:.2%}")
```

```
--- Pair tradind Perfomance in 2024 ---
 1. Cummulative return : 1.0609
 2. Sharpe ratio       : 0.53
 3. Max Drawdown       : -16.66%
```

Part III

사례로 보는 금융공학('25 봄)

사례로 보는 금융공학 실무

코스피200 변동성 조정 위클리 양매도 전략

25년도 봄학기 금융공학 특수논제 A조

김경재(20259048) / 강상묵(20259013) / 김형환(20249132) / 어환석(20259248) / 유수형(20259273)

1. 개요

본 리포트는 옵션과 변동성지수를 활용한 **변동성 조정 위클리 양매도**전략을 알아보고, **과거 데이터를 통해 구현 및 검증**하기 위해 작성되었습니다. **변동성 조정 위클리 양매도**란 ***1.*변동성 조정**, ***2.*주단위 매도** 두가지 특징을 통해 **기존 단점을 보완**한 양매도 전략으로, "매주" V-Kospi200를 이용해 행사가격 범위를 결정하고 콜/풋옵션을 매도(양매도)하게 됩니다.

전략 소개에 앞서 필요한 배경지식을 다루고, 전략 소개 및 구현, 백테스팅 및 성과를 차례로 설명하겠습니다.

2. 배경지식

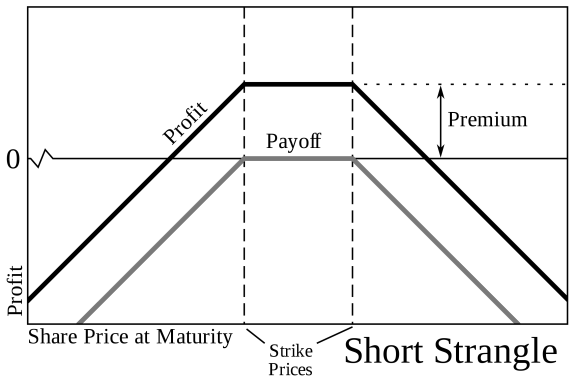
양매도 (Short strangle)

양매도란 옵션 거래전략으로, 일반적으로 **만기일이 같은 외가격(OTM) 콜/풋옵션을 매도**하는 것을 의미합니다.

OTM 옵션을 매도하므로, 만기일에 기초자산의 가격이 풋옵션의 행사가격과 콜옵션의 행사가격 사이라면 권리행사되지 않게 되고 **옵션 프리미엄(매도수익)을 그대로 얻을 수 있게 됩니다.**

반면, **양매도는 시장의 변동성이 예상과 달리 큰 경우, 큰 손실이 발생할 수 있다는 단점**도 존재합니다. **수익은 프리미엄으로 한정**되어있으나, 시황 급변에 따른 **옵션 손실에는 제한이 없어** 원금 이상의 손실이 발생할 수도 있기 때문입니다.

즉, 양매도는 높은 확률로 **안정적인 중수익을 보장하나 매우 낮은 확률로 큰 손실이 발생할 수 있는 전략**이며, **향후 시장의 변동성이 크지 않을 것이라 예측될 때 주로 사용**됩니다.



양매도 전략 예시

국내의 경우 행사가격의 상하단을 등가격(ATM) ± 5%로 설정한 양매도 ETN이 한 때 큰 인기를 끌었습니다. 대표적인 예시인 **월별 코스피200 OTM 5% 양매도 전략**의 거래방법을 살펴보면, 다음과 같습니다.

1. 현재 코스피200지수를 기준으로 ATM+5%의 콜옵션과 ATM-5%의 풋옵션을 매도 (프리미엄 수익 발생)
2. 다음달 만기일이 되면, 옵션은 청산(권리행사시 손실)되고 다시 ATM ± 5%의 콜,풋옵션을 매도

즉, **월단위로 옵션을 교체하게 되며 행사가격의 상하단은 ATM** ± 5%로 고정한 양매도 전략입니다. 중위험-중수익으로 흥행에 성공한 상품이지만, **아래와 같은 한계점**도 존재합니다.

1. 행사가격 범위가 고정되어있어 변동성 확대시 손실 가능성이 커지고, 변동성 축소시 프리미엄 수익이 크게 감소
2. 옵션 교체주기가 1개월로, 그간 지수의 하락이 누적된다면 손실이 과도하게 커질 수 있음 (유동성 리스크)

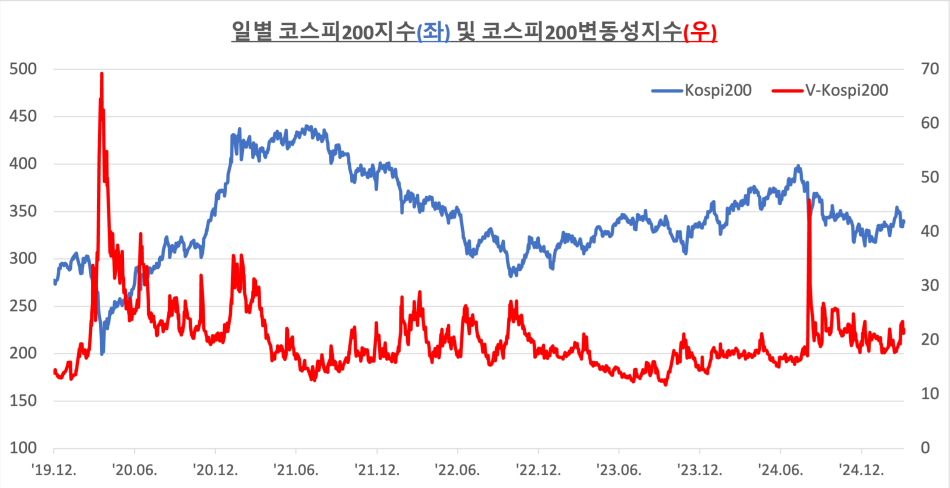
코스피200 변동성지수 (V-Kospi200 index)

코스피200 변동성 지수란 주식 시장의 **변동성을 측정하는 지표**입니다. 코스피200 옵션 가격 기반의 내재변동성을 이용하여 산출되며, 향후 30일간 시장 변동성에 대한 투자자들의 기대를 나타내는 지수입니다.

주요 특징

- **시장 심리 반영**: 투자자들의 불안 심리를 반영하며, '공포 지수(Fear Index)'라고도 불림
- **옵션 가격 기반**: 코스피200 옵션 가격 기반의 내재변동성을 통해 산출되며, 여기에는 투자자들의 기대가 반영

그래프 *1* : 일별 코스피 *200*지수 및 코스피 *200* 변동성지수



코스피200 위클리옵션

코스피200 위클리옵션이란 코스피200 지수를 기초자산으로 하는 **주간 단위의 옵션 거래 상품**을 말합니다. '19년 상장되었으며 기존의 월 단위 만기가 도래하는 옵션과는 달리, 매주 월/목요일에 만기가 도래하는 단기 옵션입니다.

주요 특징

- **세밀한 거래 지원**: 매주 월/목요일에 만기가 도래하여 단기적인 시장 변동에 대한 투자 및 위험 관리에 유리
- **다양한 투자 전략 활용**: 단기적인 시장 예측을 기반으로 다양한 투자 전략을 구사할 수 있습니다.

3. 전략 소개 및 구현

전략 개요

변동성 조정 위클리 양매도(Volatility Adjusted Weekly Short strangle, VWss) 전략이란, 기존에 널리 활용되던 **월별 OTM 5% 양매도 전략**에 아래 두가지 방법을 결합한 전략을 의미합니다.

1. 옵션의 교체주기를 "매월" -> "매주" 단위로 세분화
2. 행사가격의 범위를 고정하는 것이 아니라 매 옵션 매도시점마다 시장 변동성을 고려하여 조정

따라서, 기존과 달리 **주단위로 옵션을 교체하며, 행사가격의 범위는 교체시점에 매번 달라지게 됩니다.** 이를 위해 코스피200위클리옵션**을 사용하고, 변동성 지표는** 내재변동성 기반의 V-Kospi200지수를 사용**하여 구현할 계획입니다.

행사가격 범위 설정방법

행사가격의 상하단은 **옵션 매도시점의 "전일 V-Kospi200 종가"를 참조하여 설정**할 예정입니다. V-Kospi200는 향후 30일간 코스피200 지수의 변동성을 수치화한 지표로서, 현재시점에서 변동성을 잘 예측할 수 있는 수단이기 때문입니다.

다만, 본 전략에서 옵션 매도포지션의 유지기간은 약 7일이므로 지수 종가(연율)를 주단위 기간에 맞도록 환산하여 행사가격의 상하단을 산출하였습니다.

$$\sigma_{Target} = \frac{VKospi200_{(T-1)} \times \sqrt{Calendar\ days}}{\sqrt{365}}$$

$$Call\ strike = Kospi200_{(T)} \times (1 + \sigma_{Target})\ rounded\ up\ to\ 2.5pt$$

$$Put\ strike = Kospi200_{(T)} \times (1 - \sigma_{Target})\ rounded\ down\ to\ 2.5pt$$

이렇게 행사가격을 설정하면 V-Kospi200가 15pt일 때 월환산 변동성이 약 5%(주환산 2.5%)가 됩니다. 즉,

- 시장의 예상 변동성(V-Kospi200)이 연 15% 이상 -> 행사가격 범위를 기존보다 넓게 설정하여 손실 가능성 축소
- 시장의 예상 변동성이 연 15% 미만 -> 행사가격 범위를 기존보다 좁게 설정하여 프리미엄 수익 극대화

> 본 전략은 위클리옵션을 사용하므로 향후 1주일 변동성이 필요합니다. 따라서, 30일 변동성을 나타내는 V-Kospi200은 만기 mismatch가 있지만, 단기간의 예측치에는 큰 차이가 없고 이외의 대안(e.g. 7-day VIX)이 없어 V-Kospi200를 그대로 사용하였습니다.

포트폴리오 구성 방법

먼저, 편의를 위해 세금/수수료/호가스프레드 등의 거래비용은 없으며 종가에 원하는 수량만큼 거래할 수 있는 **완전자본시장을 가정**하도록 하겠습니다. 전략 구현을 위한 포트폴리오(투자원금 100억원) 구성 과정은 아래와 같습니다.

1. 전일 V-Kospi200 지수를 주단위로 환산하여 예상변동성($\hat{\sigma} = (VKospi200 \times \sqrt{day})/\sqrt{365}$) 산출
2. 예상변동성을 당일 Kospi200지수에 적용하여 행사가격 상하단(ATM $\pm\hat{\sigma}$을 2.5pt 단위로 올림/내림) 산출
3. 당일 Kospi200지수 및 승수(25만)를 적용, 옵션 매도수량($Q_{sell} = nominal/(kospi200 \times multiplier)$) 산출
4. 행사가격 상단 콜옵션, 하단 풋옵션 매도 ($Premium = (callprice + putprice) \times Q_{sell} \times multiplier$)
5. 원금과 매도수익을 다음주 만기일까지 MMF에 투자 ($Interest = (nominal + Premium) \times MMF \times \frac{day}{365}$)
6. 다음주 만기일이 되면, 권리행사 손실을 포함하여 최종손익 산출($Revenue = Premium + Interest - Exercise$)
    - Kospi200 〉 행사가격 상단, 콜옵션에서 손실 발생($Exercise = (Kospi200 - Strike_{call}) \times multiplier$)
    - Kospi200 〈 행사가격 하단, 풋옵션에서 손실 발생($Exercise = (Strike_{put} - Kospi200) \times multiplier$)
    - 이외의 경우, 권리행사되지 않아 옵션 포지션 청산($Exercise = 0$)
7. 주간 최종손익을 정산($nominal_{new} = nominal_{old} + Revenue$)하고, 투자종료시점까지 1. ~ 6. 과정을 반복

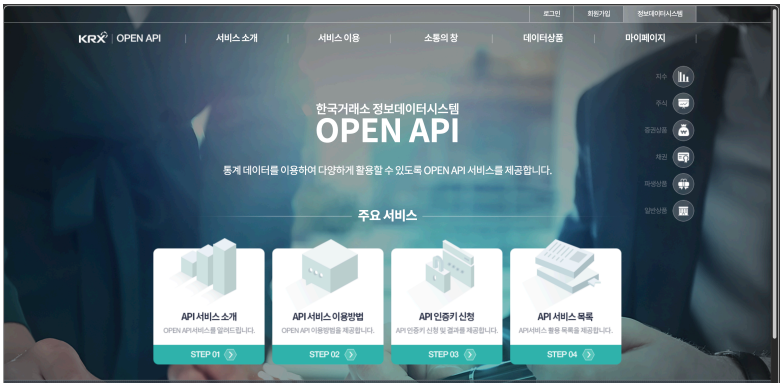
3. Backtesting

데이터 수집 및 전처리, 포트폴리오 구현

과거 5개년(2020~2024) 코스피200 등의 지수/옵션 가격, 금리를 수집하였으며, 출처는 아래와 같습니다.

`한국거래소 정보데이터시스템(data.krx.co.kr) : 코스피200 및 V-Kospi200지수`
`한국거래소 OpenAPI(openapi.krx.co.kr) : 코스피200옵션 및 위클리옵션 종목별 가격`
`한국은행 경제통계시스템(ecos.bok.or.kr) : 일별 MMF(7일) 금리`

> 한국거래소 OpenAPI를 활용한 데이터 수집 예시



API 호출시 json 데이터를 수집할 수 있으며, 이를 Dataframe(python) 및 tibble(r)로 변환하여 활용하였습니다.

```
import requests; import json
url = 'http://data-dbg.krx.co.kr/svc/sample/apis/drv/opt_bydd_trd?basDd=20250312'
headers = {'AUTH_KEY': '74D1B99DFBF345BBA3FB4476510A4BED4C78D13A'}
res = requests.get(url=url, headers=headers); res.text
```

```
'{"OutBlock_1":[{"BAS_DD":"20250312","PROD_NM":"코스피200 옵션","RGHT_TP_NM":"CALL","ISU_CD":"20
```

수집한 데이터는 **R을 이용하여 전처리**하였으며, 두개의 데이터셋으로 요약하였습니다.

1. **포트폴리오 기본 정보** : 옵션 만기일(매주), 옵션 보유일, MMF금리, Kospi200, 전일 V-K200, 거래대상옵션 정보

BAS_DD	VOL	DIFF_DAY	MMF	KOSPI200	LAG_VK200	PRIOR	TARGET_C_TK	TARGET_P_K
20241219	0.0248719	7	0.0334	322.38	17.96	24124	332.5	312.5
20241212	0.0285279	7	0.0335	329.04	20.60	24123	340.0	317.5
20241205	0.0295527	7	0.0341	323.87	21.34	24122	335.0	312.5
20241128	0.0252043	7	0.0340	331.45	18.20	24121	340.0	322.5
20241121	0.0276001	7	0.0344	329.49	19.93	24114	340.0	320.0
20241114	0.0344274	7	0.0340	317.70	24.86	24113	330.0	305.0

2. **거래대상옵션 정보** : 포트폴리오 기본 정보에 대응되는 거래대상옵션의 종가, 거래량(0인 경우 제외)

BAS_DD	PRICE_TARGET_C_PRICE	PRICE_TARGET_P_PRICE	TRDVOL_TARGET_C_K	TRDVOL_TARGET_P_K
20241219	0.29	0.74	343	137
20241212	0.42	0.78	190	196
20241205	0.44	0.91	65452	51417
20241128	0.34	0.43	288	201
20241121	0.37	0.53	115	201
20241114	0.43	0.74	329	165

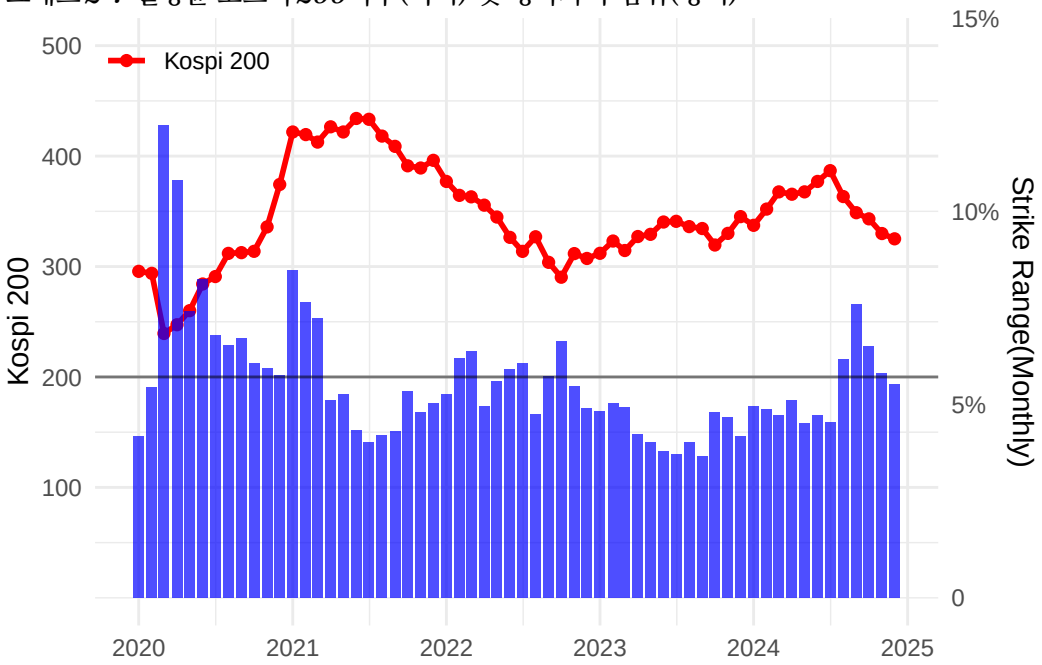
이를 통해 매도수량, 프리미엄, 이자수익, 권리행사손실 등을 산출하여 **포트폴리오를 구현**하였습니다.

BAS_DD	VOL	SELL_AMT	PREMIUM	INTEREST	EXERCISE	REVENUE	RATE
20200102	0.0203434	137.7648	29963837	2789154	0	32752991	0.0032753
20200109	0.0221160	135.8650	20040080	2844044	9510547	13373578	0.0013374
20200116	0.0185154	132.1091	18825550	2824485	0	21650035	0.0021650
20200123	0.0197895	132.3058	23815037	2787444	219296795	-192694314	-0.0192694
20200130	0.0235286	138.7107	45080972	2793358	109234664	-61360333	-0.0061360
20200206	0.0252458	133.0451	46565774	2774504	0	49340278	0.0049340

❗ 변동성 조정 방법에 따른 행사가격 범위 추이(과거 5년)

V-Kospi200과 연동한 행사가격의 범위는 지난 5년간 **1.6% ~ 8.7%까지 광범위하게 형성**되었습니다.
코스피200이 급등락하는 경우 범위가 넓게 형성되고 보합장에서는 좁게 형성되는 추이를 확인할 수 있으며, 이를 월환산($\times\sqrt{4}$)하여 기존 5%와 비교하면 **지수의 상황에 따라 유동적으로 조절되고 있음**을 의미합니다.

그래프 *2* : 월평균 코스피 *200*지수(적색) 및 행사가격 범위(청색)



Backtesting 성과

지난 5년간(2020~2024) VW양매도와 코스피200지수 / OTM 5% 양매도를 비교해보았습니다.

표*1 :* 연도별 변동성 조정 위클리 양매도 포트폴리오 및 코스피*200*지수 성과

YEAR	Expiry	NoExercise	Premium	Interest	Loss	Revenue	Return	Return_K200
2020	50	0.74	4385	184	5338	-769	-0.04	0.33
2021	47	0.91	2974	142	728	2387	0.12	0.01
2022	52	0.79	3304	421	4152	-427	-0.03	-0.26
2023	52	0.81	2456	732	1758	1430	0.08	0.23
2024	49	0.80	3697	719	3454	962	0.05	-0.11

표*2 :* 연도별 일반 양매도 포트폴리오 및 코스피*200*지수 성과

YEAR	Expiry	NoExercise	Premium	Interest	Loss	Revenue	Return	Return_K200
2020	12	0.58	16507	847	37851	-20497	-0.23	0.33
2021	12	1.00	9887	605	0	10492	0.13	0.01
2022	12	0.50	10207	1819	15098	-3071	-0.04	-0.26
2023	12	0.83	4591	3176	884	6883	0.09	0.23
2024	12	0.75	9850	3072	9589	3333	0.04	-0.11

YEAR : 산출대상 연도 / **Expiry** : 옵션만기 횟수 (프리미엄 수익 발생 횟수)
NoExercise : 옵션만기일에 행사되지 않은 비율 (손실이 발생하지 않은 거래일 비율)
Premium : 평균 옵션 프리미엄 수익 (만원) / **Interest** : 원금 및 프리미엄에서 발생한 평균 이자수익 (만원)
Loss : 옵션권리행사로 인한 평균 손실 (미행사 포함) / **Revenue** : 평균 이익 (Premium + Interest - Loss)
Return : 포트폴리오의 연환산 수익률 / **Return_K200** : 코스피200지수의 연환산 수익률

변동성 조정 위클리 양매도 포트폴리오의 주요 성과는 다음과 같습니다.

1. 옵션 매도주기 축소(월간 → 주간) : **현금흐름 개선 및 프리미엄 수익 극대화**

- VW양매도 포트폴리오는 연평균 50회의 수익이 발생하는 반면, 일반 양매도 포트폴리오는 12회(월1회) 발생.
- 1회 발생 수익은 4배 미만으로 감소하여 평균 수익이 증가하였고, 수익주기가 짧아지면서 현금유동성 개선

2. 행사가격 범위 조정(5% → σ 연동) : **포트폴리오 안정성 증가 및 리스크 축소**

- 일반 양매도 전략의 손실발생비율(권리행사비율)은 시장 상황에 따라 0~50%까지 큰 폭으로 변동
- 반면, 본 포트폴리오는 행사가격 범위를 변동성 수준에 조정하므로 비율이 10~30%수준으로 안정화
- 손실에 대한 예측가능성 및 포트폴리오 변동성이 개선되었으며 1회 손실도 감내 가능한 수준으로 축소

3. **소결 : VW양매도 전략은 수익률, 리스크 측면에서 코스피200지수 및 일반 양매도 전략을 Outperform**

그래프 *3, 4* : 지난 *5*년 및 *3*년간 코스피*200*지수, ***VW***양매도, *5%*양매도 누적수익률 및 초과수익

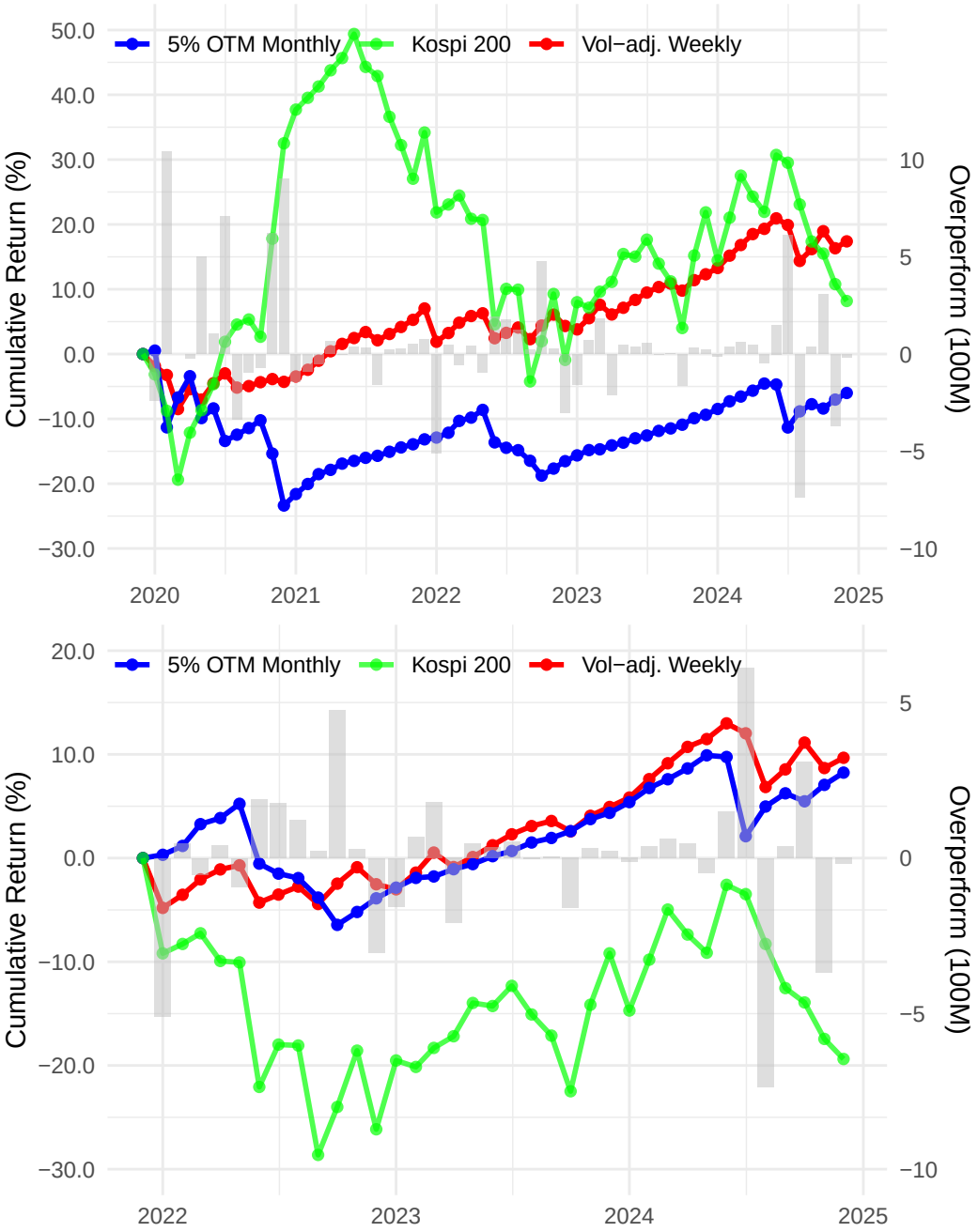


표 *3* : 연도별 ***VW***양매도, 일반 양매도, 코스피*200*지수의 수익률 및 변동성(월간수익률의 연환산)

YEAR	Return_main	Return_sub	Return_K200	Vol_main	Vol_sub	Vol_K200
2020	-0.04	-0.23	0.33	0.08	0.19	0.27
2021	0.12	0.13	0.01	0.03	0.02	0.11
2022	-0.03	-0.04	-0.26	0.08	0.08	0.25
2023	0.08	0.09	0.23	0.04	0.01	0.17
2024	0.05	0.04	-0.11	0.07	0.08	0.16

그래프와 표를 통해 **VW양매도 전략**이 타 전략 대비 **안정적이고 높은 수익을 실현**하였음을 확인할 수 있었습니다.

한계점

그러나, **행사가격 범위가 전일 V-Kospi200 지수에 따라 결정**되므로 옵션 매도일 및 보유기간 중 V-Kospi200 지수가 급등락하는 경우 **시장 변동성을 적절히 반영되지 않을 가능성**이 있습니다.

- 당일 장중 지수를 사용할 수 있겠으나 종가보다 신뢰성이 높다고 보기 어렵고, 당일 종가는 매도시점에서 확인 불가
- 보유기간 중 V-Kospi200의 변동에 따라 옵션 행사가격을 리밸런싱하면 보다 정확히 변동성을 반영할 수 있으나, 잦은 거래로 인해 수반되는 비용이 급증
- 시장 변동성이 적절히 반영되지 않는 경우, 프리미엄 수익성이 낮아지고 옵션 권리행사 위험이 증가할 수 있음

또한, **실제로 VW양매도 포트폴리오를 운영**한다면 **거래비용 및 유동성 등의 한계점**으로 인해 보완해야할 점이 있으며 이 과정에서 초과수익률 등의 **성과가 다소 희석**될 것으로 예상됩니다.

- 수수료/유동성 등으로 인해 자주 옵션을 매도하는 전략 특성상 거래비용 상승이 수반됨
- 월물 옵션 대비 위클리옵션의 유동성이 낮아, 변동성 확대 국면에서 원하는 위클리옵션의 거래가 없을 수 있음

4. 포트폴리오 검증 ('25.3.13 ~ 4.10)

포트폴리오의 성과 검증을 위해 **'25년 3월 옵션만기일부터 1개월간의 성과를 측정**해보겠습니다.

Backtesting과 동일한 방식으로 진행하였으며, 날짜 등 일부를 제외하고 동일 코드를 재사용하였습니다.

표*4 : 검증기간('25.3.13 ~ 4.10) VW*양매도 및 일반 양매도 전략 성과

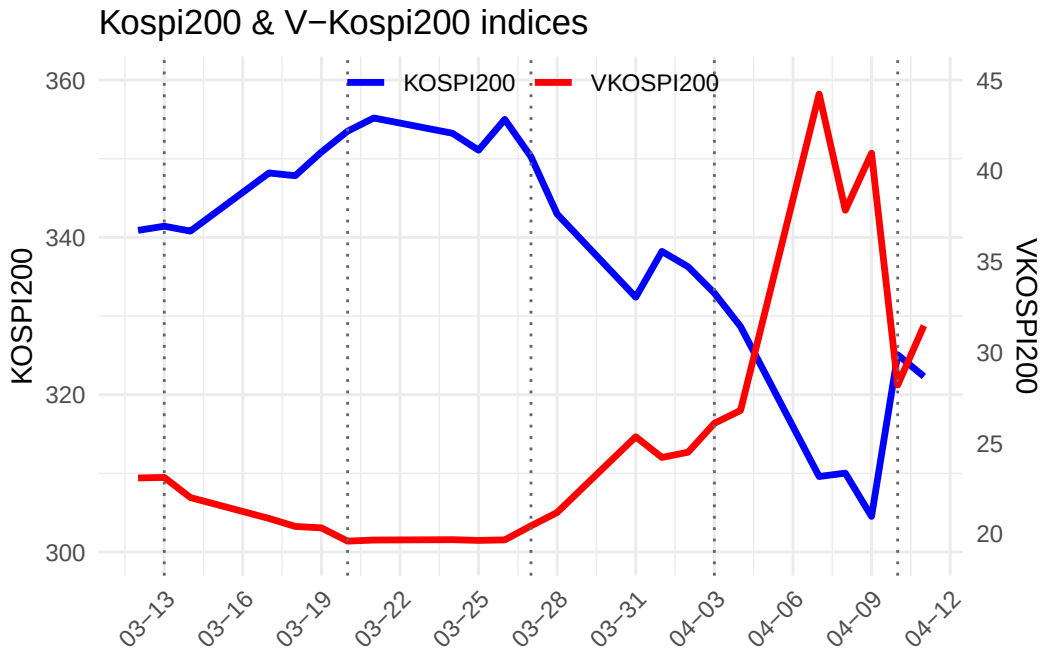
BAS_DD	Group	Premium	Interest	Loss	Revenue	Return
20250313	Monthly	12653	2400	0	15053	0.02
20250313	Vol-adj. Weekly	5624	596	2929	3291	0.00
20250320	Vol-adj. Weekly	2687	587	0	3274	0.00
20250327	Vol-adj. Weekly	4283	578	20214	-15353	-0.02
20250403	Vol-adj. Weekly	11594	578	0	12173	0.01
81001363	Vol-adj. Weekly Sum	24188	2338	23142	3384	0.00

검증기간 중 VW양매도 전략은 옵션을 4회 매도하였고, 일반 양매도는 1회 매도하였습니다.

VW양매도 전략의 **프리미엄이 4회 발생하면서 수익성은 개선**되었지만, 세번째 매도시기에 **큰 손실이 발생하면서 순이익이 악화**되었습니다. 벡테스팅 결과와 비교할 때 **상반된 결과**입니다.

그 **원인은 최근 트럼프 행정부의 관세 부과 정책의 영향**으로, 증시가 이례적으로 급등락한 데에서 찾을 수 있었습니다.

그래프 *5 : 검증기간('25.3.13 ~ 4.10) Kospi200 및 V-Kospi200 지수 추이*



먼저, **3.27일 예상변동성(V-K200)이 낮아 행사가격 범위가 좁게 형성**되어 VW양매도 전략이 실행되었고, 이후 관세 정책 영향으로 지수가 급락하면서 **큰 손실이 발생**하게 되었습니다.

반면 **월별 전략은 만기 직전에 관세가 연기**되면서 **증시가 회복**하였고(304 〉 325), 손실을 피하게 되었습니다.

다소 아쉬운 결과이나, **VW양매도 전략의 한계점과 특수한 상황에서는 오히려 불리하다는 점**을 알 수 있었습니다.

1. VW양매도 전략은 **시장 변동성이 적절히 반영되지 않을 가능성**이 있고, 이 경우 예기치 못한 손실이 발생할 수 있음
2. **옵션 만기가 짧은 것은 일반적**으로 수익성, 유동성 등에서 **장점**이 있으나, **증시가 급락 후 회복하는 상황에서는 만기가 긴 옵션이 유리**할 수 있음

5. Appendix : Python, R code

Pyhon code : 데이터 수집 및 전처리

```python
import requests
import json
import pandas as pd
import aiohttp
import asyncio

# KRX OpenAPI 예시
url = 'http://data-dbg.krx.co.kr/svc/sample/apis/drv/opt_bydd_trd?basDd=20250312'
headers = {'AUTH_KEY': '74D1B99DFBF345BBA3FB4476510A4BED4C78D13A'}
res = requests.get(url=url, headers=headers)
res.text

# KRX OpenAPI를 이용한 옵션 데이터 수집 (네트워크 병렬 처리)
idx_data = pd.read_csv('data/idx_data.csv')
url = 'http://data-dbg.krx.co.kr/svc/apis/drv/opt_bydd_trd?basDd='
key = 'BDFA640BBCE84C4B8A465EA024D50D6F3FD909FF'

async def fetch(session, url, bas_dd):
    async with session.get(url + bas_dd, headers={'AUTH_KEY': key}) as response:
        return await response.json()

async def fetch_all(bas_dd_list, url):
    async with aiohttp.ClientSession() as session:
        tasks = [fetch(session, url, str(bas_dd)) for bas_dd in bas_dd_list]
        return await asyncio.gather(*tasks)

async def main():
    bas_dd_list = idx_data['BAS_DD'].astype(str).tolist()
    responses = await fetch_all(bas_dd_list, url)

    data_list = [pd.json_normalize(res['OutBlock_1']) for res in responses if 'OutBlock_1' in
    options = pd.concat(data_list, axis=0, ignore_index=True)

    # CSV 저장
    options.to_csv("options_data.csv", encoding="utf-8-sig", index=False)
```

```
    print("complete")
```

R code 1 : 데이터 가공, Backtesting

```
rm(list=ls())
library(tidyverse)
library(knitr)
library(patchwork)

# 1. 원본데이터 준비
options_raw <- read_csv("data/options_data.csv") %>% tibble()
idx_raw <- read_csv("data/idx_data.csv") %>% tibble()
mmf_raw <- read_csv("data/mmf.csv") %>% tibble()

# 2-1. 데이터 전처리 : KRX openAPI 옵션데이터 전처리
options <- options_raw %>%
  # K200, WK200 이외의 옵션 제외
  filter(substr(PROD_NM,1,3)=="코스피") %>%
  # 종가가 없는 경우 익일 기준가격(이론가격) 사용
  mutate(PRICE = as.double(if_else(TDD_CLSPRC=="-",NXTDD_BAS_PRC,TDD_CLSPRC))) %>%
  drop_na(PRICE) %>%
  select(BAS_DD,ISU_NM,PRICE,ACC_TRDVOL) %>%
  separate(ISU_NM, into=c("PROD","RGHT","EXP","EXER_PRC"), sep=' ') %>%
  mutate(EXER_PRC=as.double(EXER_PRC),
         EXPMM=if_else(PROD=="코스피200",substr(EXP,3,6),substr(EXP,1,4)),
         EXPWW=if_else(PROD=="코스피200",2,as.integer(substr(EXP,6,6)))) %>%
  filter(EXER_PRC>min(idx_raw$KOSPI200)*0.85,
         EXER_PRC<max(idx_raw$KOSPI200)*1.15,
         PROD!="코스피위클리M") %>% # 월요일 만기 위클리옵션 제외
  mutate(PRIOR=as.integer(EXPMM)*10+EXPWW) # 만기가 짧은 순으로 우선순위 부여

# 2-2. 데이터 전처리 : 옵션 만기일 데이터 생성
target_date <- options %>%
  distinct(.,BAS_DD, PRIOR) %>%
  arrange(PRIOR, desc(BAS_DD)) %>%
  group_by(PRIOR) %>%
  slice(1) %>%
  ungroup() %>%
```

```
  distinct(BAS_DD) %>%
  arrange(desc(BAS_DD)) %>%
  filter(BAS_DD<20241231, BAS_DD>20200101)

# 2-3. 데이터 전처리 : KRX 정보데이터시스템 K200 및 V-K200지수 전처리
idx_data <- idx_raw %>%
  arrange(BAS_DD) %>%
  mutate(LAG_K200=lag(KOSPI200),
         LAG_VK200=lag(VKOSPI200))

K200_portfolio=idx_data %>%
  left_join(mmf_raw, by="BAS_DD") %>%
  mutate(RATE=(KOSPI200-LAG_K200)/LAG_K200,
         YEAR = substr(BAS_DD, 1, 4),
         YM = ym(substr(BAS_DD, 1, 6))) %>%
  group_by(YEAR, YM) %>%
  summarise(RATE_K200 = if_else(is.na(prod(1 + RATE)),0,prod(1 + RATE)),
            MMF = mean(MMF)/100,
            .groups = "drop") %>%
  ungroup()

# 3-1. 포트폴리오 구성 : VW양매도 포트폴리오 기초정보 생성
target_info <- target_date %>%
  left_join(options, by="BAS_DD") %>%
  distinct(BAS_DD,PRIOR) %>%
  arrange(desc(BAS_DD),PRIOR) %>%
  group_by(BAS_DD) %>%
  # 만기가 도래하는 옵션을 제외하고 우선순위가 높은 옵션 선택
  slice(2) %>%
  ungroup() %>%
  arrange(desc(BAS_DD)) %>%
  left_join(idx_data, by="BAS_DD") %>%
  left_join(mmf_raw, by="BAS_DD") %>%
  # 옵션 보유기간 및 단기금리(MMF) 계산
  mutate(DIFF_DAY=as.integer(ymd(lag(BAS_DD))-ymd(BAS_DD)),
         MMF=MMF*0.01) %>%
  # 옵션 보유기간에 해당하는 시장기대변동성 계산(V-K200활용)
  mutate(VOL=LAG_VK200*sqrt(DIFF_DAY)/sqrt(365)/100) %>%
  # 변동성에 따른 옵션 행사가격 상단(2.5단위 올림) 및 하단(2.5단위 내림) 계산
```

```
  mutate(TARGET_C_K=ceiling(KOSPI200*(1+VOL)*0.4)/0.4,
         TARGET_P_K=floor(KOSPI200*(1-VOL)*0.4)/0.4) %>%
  drop_na(DIFF_DAY) %>%
  select(BAS_DD,VOL,DIFF_DAY,MMF,KOSPI200,LAG_VK200,PRIOR,TARGET_C_K,TARGET_P_K)

# 3-2. 포트폴리오 구성 : VW양매도 포트폴리오 거래대상 옵션 정보 생성
target_options <- target_info %>%
  select(BAS_DD, PRIOR, TARGET_C_K, TARGET_P_K) %>%
  pivot_longer(cols=c("TARGET_C_K","TARGET_P_K"),names_to = "GRP", values_to = "EXER_PRC") %>%
  mutate(RGHT=substr(GRP,8,8)) %>%
  left_join(options,by=c("BAS_DD","PRIOR","RGHT","EXER_PRC")) %>%
  select(BAS_DD,GRP,PRICE,ACC_TRDVOL) %>%
  pivot_wider(names_from = "GRP", values_from = c("PRICE","ACC_TRDVOL"))

# 원금 100억원 가정
cash = 10000*10000*100

# 4. 포트폴리오 구현 : VW양매도 전략을 구현하여 일자별 손익 계산
portfolio <- target_info %>%
  left_join(target_options,by="BAS_DD") %>%
  arrange(BAS_DD) %>%
  # 원금에 따른 옵션 매도수량 계산
  mutate(SELL_AMT=cash/250000/KOSPI200) %>%
  # 옵션 프리미엄(매도수익)) 계산
  mutate(PREMIUM=SELL_AMT*(PRICE_TARGET_C_K+PRICE_TARGET_P_K)*250000) %>%
  # 원금 및 프리미엄의 MMF 이자수익 및 권리행사손실 계산
  mutate(INTEREST=(cash+replace_na(PREMIUM,0))*MMF*DIFF_DAY/365,
         EXERCISE=(if_else(lead(KOSPI200)-TARGET_C_K>0,lead(KOSPI200)-TARGET_C_K,0)+
                     if_else(TARGET_P_K-lead(KOSPI200)>0,TARGET_P_K-lead(KOSPI200),0))*250000*
  # 주단위 포트폴리오 손익 계산(원금 100억 가정, 당일 투자시 다음주 실현손익을 의미)
  mutate(REVENUE=PREMIUM+INTEREST-EXERCISE) %>%
  # 거래대상 옵션이 상장되어있지 않은 경우, MMF수익만 고려
  mutate(REVENUE=if_else(is.na(REVENUE),INTEREST,REVENUE)) %>%
  mutate(RATE=REVENUE/cash)  # 주단위 포트폴리오 수익률

# 5. 비교군 포트폴리오 생성 : 월별 5% OTM 양매도 전략
options2 <- options %>%
  filter(PROD=="코스피200")
```

```r
# 옵션 만기일(매달) 생성
target_date2 <- options2 %>%
  distinct(.,BAS_DD, PRIOR) %>%
  arrange(PRIOR, desc(BAS_DD)) %>%
  group_by(PRIOR) %>%
  slice(1) %>%
  ungroup() %>%
  distinct(BAS_DD) %>%
  arrange(desc(BAS_DD)) %>%
  filter(BAS_DD<20241231, BAS_DD>20200101)

# 일반 양매도 포트폴리오 기본 정보 생성
target_info2 <- target_date2 %>%
  left_join(options2, by="BAS_DD") %>%
  distinct(BAS_DD,PRIOR) %>%
  arrange(desc(BAS_DD),PRIOR) %>%
  group_by(BAS_DD) %>%
  # 만기가 도래하는 옵션을 제외하고 우선순위가 높은 옵션 선택
  slice(2) %>%
  ungroup() %>%
  arrange(desc(BAS_DD)) %>%
  left_join(idx_data, by="BAS_DD") %>%
  left_join(mmf_raw, by="BAS_DD") %>%
  # 옵션 보유기간 및 단기금리(MMF) 계산
  mutate(DIFF_DAY=as.integer(ymd(lag(BAS_DD))-ymd(BAS_DD)),
         MMF=MMF*0.01,
         VOL=0.05) %>%
  # 변동성에 따른 옵션 행사가격 상단(2.5단위 올림) 및 하단(2.5단위 내림) 계산
  mutate(TARGET_C_K=ceiling(KOSPI200*(1+VOL)*0.4)/0.4,
         TARGET_P_K=floor(KOSPI200*(1-VOL)*0.4)/0.4) %>%
  mutate(DIFF_DAY=if_else(is.na(DIFF_DAY),28,DIFF_DAY)) %>%
  select(BAS_DD,VOL,DIFF_DAY,MMF,KOSPI200,LAG_VK200,PRIOR,TARGET_C_K,TARGET_P_K)

# 일반 양매도 포트폴리오 거래대상 옵션
target_options2 <- target_info2 %>%
  select(BAS_DD, PRIOR, TARGET_C_K, TARGET_P_K) %>%
  pivot_longer(cols=c("TARGET_C_K","TARGET_P_K"),names_to = "GRP", values_to = "EXER_PRC") %>%
  mutate(RGHT=substr(GRP,8,8)) %>%
  left_join(options2,by=c("BAS_DD","PRIOR","RGHT","EXER_PRC")) %>%
```

```
  select(BAS_DD,GRP,PRICE,ACC_TRDVOL) %>%
  pivot_wider(names_from = "GRP", values_from = c("PRICE","ACC_TRDVOL"))

# 일반 양매도 포트폴리오 구현
portfolio2 <- target_info2 %>%
  left_join(target_options2,by="BAS_DD") %>%
  arrange(BAS_DD) %>%
  # 원금에 따른 옵션 매도수량 계산
  mutate(SELL_AMT=cash/250000/KOSPI200) %>%
  # 옵션 프리미엄(매도수익)) 계산
  mutate(PREMIUM=SELL_AMT*(PRICE_TARGET_C_K+PRICE_TARGET_P_K)*250000) %>%
  # 원금 및 프리미엄의 MMF 이자수익 및 권리행사손실 계산
  mutate(INTEREST=(cash+replace_na(PREMIUM,0))*MMF*DIFF_DAY/365,
         EXERCISE=(if_else(lead(KOSPI200)-TARGET_C_K>0,lead(KOSPI200)-TARGET_C_K,0)+
                     if_else(TARGET_P_K-lead(KOSPI200)>0,TARGET_P_K-lead(KOSPI200),0))*250000*
  mutate(EXERCISE=if_else(is.na(EXERCISE),0,EXERCISE)) %>%
  # 주단위 포트폴리오 손익 계산(원금 100억 가정, 당일 투자시 다음주 실현손익을 의미)
  mutate(REVENUE=PREMIUM+INTEREST-EXERCISE) %>%
  # 거래대상 옵션이 상장되어있지 않은 경우, MMF수익만 고려
  mutate(REVENUE=if_else(is.na(REVENUE),INTEREST,REVENUE)) %>%
  mutate(RATE=REVENUE/cash) # 주단위 포트폴리오 수익률

# 6. 성과분석 : VW양매도 포트폴리오
performance <- portfolio %>%
  drop_na(PREMIUM, EXERCISE) %>%
  mutate(YEAR = substr(BAS_DD, 1, 4),
         YM = ym(substr(BAS_DD, 1, 6))) %>%
  group_by(YEAR, YM) %>%
  summarise(PREMIUM = sum(PREMIUM),
            INTEREST = sum(INTEREST),
            EXERCISE = sum(EXERCISE),
            REVENUE = sum(REVENUE),
            RATE = prod(1 + RATE),
            .groups = "drop") %>%
  ungroup()

# 6. 성과분석 : 일반 양매도 포트폴리오
performance2 <- portfolio2 %>%
  drop_na(PREMIUM, EXERCISE) %>%
```

```
  mutate(YEAR = substr(BAS_DD, 1, 4),
         YM = ym(substr(BAS_DD, 1, 6))) %>%
  group_by(YEAR, YM) %>%
  summarise(PREMIUM = sum(PREMIUM),
            INTEREST = sum(INTEREST),
            EXERCISE = sum(EXERCISE),
            REVENUE = sum(REVENUE),
            RATE = prod(1 + RATE),
            .groups = "drop") %>%
  ungroup()

# 시각화 등
graph1_1 <- performance %>%
  arrange(YM) %>%
  left_join(K200_portfolio,by=c("YEAR","YM")) %>%
  mutate(CUM_RATE = cumprod(RATE) * 100 - 100,
         CUM_RATE_K200 = cumprod(RATE_K200) * 100 - 100) %>%
  bind_rows(tibble(YM=ym(201912),CUM_RATE=0,CUM_RATE_K200=0))


graph1_2 <- performance2 %>%
  arrange(YM) %>%
  left_join(K200_portfolio,by=c("YEAR","YM")) %>%
  mutate(CUM_RATE = cumprod(RATE) * 100 - 100,
         CUM_RATE_K200 = cumprod(RATE_K200) * 100 - 100) %>%
  bind_rows(tibble(YM=ym(201912),CUM_RATE=0,CUM_RATE_K200=0))

graph1_3 <- graph1_1 %>%
  inner_join(graph1_2, by = "YM", suffix = c("_1", "_2")) %>%
  mutate(REVENUE_DIFF = (REVENUE_1 - REVENUE_2) / 1e8)

graph2_1 <- performance %>%
  arrange(YM) %>%
  left_join(K200_portfolio,by=c("YEAR","YM")) %>%
  filter(as.integer(YEAR) > 2021) %>%
  mutate(CUM_RATE = cumprod(RATE) * 100 - 100,
         CUM_RATE_K200 = cumprod(RATE_K200) * 100 - 100) %>%
  bind_rows(tibble(YM=ym(202112),CUM_RATE=0,CUM_RATE_K200=0))
```

```r
graph2_2 <- performance2 %>%
  arrange(YM) %>%
  left_join(K200_portfolio,by=c("YEAR","YM")) %>%
  filter(as.integer(YEAR) > 2021) %>%
  mutate(CUM_RATE = cumprod(RATE) * 100 - 100,
         CUM_RATE_K200 = cumprod(RATE_K200) * 100 - 100) %>%
  bind_rows(tibble(YM=ym(202112),CUM_RATE=0,CUM_RATE_K200=0))

graph2_3 <- graph2_1 %>%
  inner_join(graph2_2, by = "YM", suffix = c("_1", "_2")) %>%
  mutate(REVENUE_DIFF = (REVENUE_1 - REVENUE_2) / 1e8)

graph3_1 <- portfolio %>%
  mutate(YM = ym(substr(BAS_DD, 1, 6))) %>%
  group_by(YM) %>%
  summarise(TargetVol=mean(VOL),
            Kospi200=mean(KOSPI200)) %>% ungroup()

graph1=ggplot() +
  geom_line(data = graph1_1, aes(x = YM, y = CUM_RATE, color = "Vol-adj. Weekly"), size = 1) +
  geom_point(data = graph1_1, aes(x = YM, y = CUM_RATE, color = "Vol-adj. Weekly"), shape = 16
  geom_line(data = graph1_2, aes(x = YM, y = CUM_RATE, color = "5% OTM Monthly"), size = 1) +
  geom_point(data = graph1_2, aes(x = YM, y = CUM_RATE, color = "5% OTM Monthly"), shape = 16,
  geom_line(data = graph1_1, aes(x = YM, y = CUM_RATE_K200, color = "Kospi 200"), size = 1, al
  geom_point(data = graph1_1, aes(x = YM, y = CUM_RATE_K200, color = "Kospi 200"), shape = 16,
  geom_bar(data = graph1_3, aes(x = YM, y = REVENUE_DIFF * 3),
           stat = "identity", alpha = 0.5, fill = "gray") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  scale_y_continuous(limits = c(-30, 50), breaks = seq(-30, 50, 10), labels = scales::number_f
                     sec.axis = sec_axis(~ . / 3, name = "Overperform (100M)", breaks = c(-10,
  scale_color_manual(values = c("Vol-adj. Weekly" = "red", "5% OTM Monthly" = "blue", "Kospi 2
  labs(x = NULL, y = "Cumulative Return (%)", color = "") +
  theme_minimal() +
  theme(legend.position = c(0.4, 0.93), legend.direction = "horizontal")

graph2=ggplot() +
  geom_line(data = graph2_1, aes(x = YM, y = CUM_RATE, color = "Vol-adj. Weekly"), size = 1) +
  geom_point(data = graph2_1, aes(x = YM, y = CUM_RATE, color = "Vol-adj. Weekly"), shape = 16
  geom_line(data = graph2_2, aes(x = YM, y = CUM_RATE, color = "5% OTM Monthly"), size = 1) +
```

```
  geom_point(data = graph2_2, aes(x = YM, y = CUM_RATE, color = "5% OTM Monthly"), shape = 16,
  geom_line(data = graph2_1, aes(x = YM, y = CUM_RATE_K200, color = "Kospi 200"), size = 1, al
  geom_point(data = graph2_1, aes(x = YM, y = CUM_RATE_K200, color = "Kospi 200"), shape = 16,
  geom_bar(data = graph2_3, aes(x = YM, y = REVENUE_DIFF * 3),
           stat = "identity", alpha = 0.5, fill = "gray") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  scale_y_continuous(limits = c(-30, 20), breaks = seq(-30, 30, 10), labels = scales::number_f
                     sec.axis = sec_axis(~ . / 3, name = "Overperform (100M)", breaks = c(-10,
  scale_color_manual(values = c("Vol-adj. Weekly" = "red", "5% OTM Monthly" = "blue", "Kospi 2
  labs(x = NULL, y = "Cumulative Return (%)", color = "") +
  theme_minimal() +
  theme(legend.position = c(0.4, 0.93), legend.direction = "horizontal")

graph3=ggplot() +
  geom_line(data = graph3_1, aes(x = YM, y = Kospi200, color = "Kospi 200"), size = 1) +
  geom_point(data = graph3_1, aes(x = YM, y = Kospi200, color = "Kospi 200"), shape = 16, size
  geom_bar(data = graph3_1, aes(x = YM, y = TargetVol * 7000),
           stat = "identity", alpha = 0.7, fill = "blue") +
  geom_hline(yintercept = 200, size=0.5, color = "black", alpha=0.5)+
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  scale_y_continuous(limits = c(0, 500), breaks = seq(100, 500, 100),
                     sec.axis = sec_axis(~ . / 7000, name = "Strike Range(Monthly)", breaks =
  scale_color_manual(values = c("Kospi 200" = "red")) +
  labs(x = NULL, y = "Kospi 200", color = "") +
  theme_minimal() +
  theme(legend.position = c(0.1, 0.93), legend.direction = "horizontal")

# 요약표 1 : VW포트폴리오
table1 <- portfolio %>%
  drop_na(PREMIUM, EXERCISE) %>%
  mutate(YEAR = substr(BAS_DD, 1, 4),
         CNT=1,
         NO=if_else(EXERCISE==0,1,0)) %>%
  group_by(YEAR) %>%
  summarise(Expiry = sum(CNT),
            NoExercise = round(sum(NO)/sum(CNT),2),
            Premium = round(mean(PREMIUM)/10000,0),
            Interest = round(mean(INTEREST)/10000,0),
            Loss = round(mean(EXERCISE)/10000,0),
```

```
            Revenue = round(mean(REVENUE)/10000,0),
            Return = round(prod(1 + RATE)-1,2),
            .groups = "drop") %>%
  left_join(K200_portfolio %>% group_by(YEAR) %>% summarise(Return_K200=round(prod(RATE_K200)-
            by="YEAR")

# 요약표 2 : 일반 포트폴리오
table2 <- portfolio2 %>%
  drop_na(PREMIUM, EXERCISE) %>%
  mutate(YEAR = substr(BAS_DD, 1, 4),
         CNT=1,
         NO=if_else(EXERCISE==0,1,0)) %>%
  group_by(YEAR) %>%
  summarise(Expiry = sum(CNT),
            NoExercise = round(sum(NO)/sum(CNT),2),
            Premium = round(mean(PREMIUM)/10000,0),
            Interest = round(mean(INTEREST)/10000,0),
            Loss = round(mean(EXERCISE)/10000,0),
            Revenue = round(mean(REVENUE)/10000,0),
            Return = round(prod(1 + RATE)-1,2),
            .groups = "drop") %>%
  left_join(K200_portfolio %>% group_by(YEAR) %>% summarise(Return_K200=round(prod(RATE_K200)-
            by="YEAR")

# 요약표 3 : 포트폴리오 변동성 비교
Vol1 <- graph1_1 %>% group_by(YEAR) %>% summarise(Vol=round(sd(RATE)*sqrt(12),2),
                                                  Vol_K200=round(sd(RATE_K200)*sqrt(12),2))
Vol2 <- graph1_2 %>% group_by(YEAR) %>% summarise(Vol=round(sd(RATE)*sqrt(12),2))

table3 <- table1 %>%
  select(YEAR,Return,Return_K200) %>%
  left_join(Vol1, by="YEAR") %>%
  mutate(Return_main=Return,
         Vol_main=Vol) %>%
  select(YEAR, Return_main,Vol_main,Return_K200,Vol_K200) %>%
  left_join(table2 %>%
              select(YEAR,Return) %>%
              left_join(Vol2, by="YEAR") %>%
              mutate(Return_sub=Return,
```

```
                    Vol_sub=Vol) %>%
            select(YEAR,Return_sub,Vol_sub), by="YEAR")
```

R code 2 : 전략 검증

```
rm(list=ls())
library(tidyverse)
library(knitr)
library(patchwork)
setwd("homepage/study_25spring/data/")

# 1. 원본데이터 준비
options_raw <- read_csv("options_data_test.csv") %>% tibble()
idx_raw <- read_csv("idx_data_test.csv") %>% tibble()
mmf_raw <- read_csv("mmf_test.csv") %>% tibble()

# 2-1. 데이터 전처리 : KRX openAPI 옵션데이터 전처리
options <- options_raw %>%
  # K200, WK200 이외의 옵션 제외
  filter(substr(PROD_NM,1,3)=="코스피") %>%
  # 종가가 없는 경우 익일 기준가격(이론가격) 사용
  mutate(PRICE = as.double(if_else(TDD_CLSPRC=="-",NXTDD_BAS_PRC,TDD_CLSPRC))) %>%
  drop_na(PRICE) %>%
  select(BAS_DD,ISU_NM,PRICE,ACC_TRDVOL) %>%
  separate(ISU_NM, into=c("PROD","RGHT","EXP","EXER_PRC"), sep=' ') %>%
  mutate(EXER_PRC=as.double(EXER_PRC),
         EXPMM=if_else(PROD=="코스피200",substr(EXP,3,6),substr(EXP,1,4)),
         EXPWW=if_else(PROD=="코스피200",2,as.integer(substr(EXP,6,6)))) %>%
  filter(EXER_PRC>min(idx_raw$KOSPI200)*0.85,
         EXER_PRC<max(idx_raw$KOSPI200)*1.15,
         PROD!="코스피위클리M") %>% # 월요일 만기 위클리옵션 제외
  mutate(PRIOR=as.integer(EXPMM)*10+EXPWW) # 만기가 짧은 순으로 우선순위 부여

# 2-2. 데이터 전처리 : 옵션 만기일 데이터 생성
target_date <- options %>%
  distinct(.,BAS_DD, PRIOR) %>%
  arrange(PRIOR, desc(BAS_DD)) %>%
  group_by(PRIOR) %>%
```

```
  slice(1) %>%
  ungroup() %>%
  distinct(BAS_DD) %>%
  arrange(desc(BAS_DD)) %>%
  filter(BAS_DD<20250411)

# 2-3. 데이터 전처리 : KRX 정보데이터시스템 K200 및 V-K200지수 전처리
idx_data <- idx_raw %>%
  arrange(BAS_DD) %>%
  mutate(LAG_K200=lag(KOSPI200),
         LAG_VK200=lag(VKOSPI200))

K200_portfolio=idx_data %>%
  left_join(mmf_raw, by="BAS_DD") %>%
  mutate(RATE=(KOSPI200-LAG_K200)/LAG_K200,
         YEAR = substr(BAS_DD, 1, 4),
         YM = ym(substr(BAS_DD, 1, 6))) %>%
  group_by(YEAR) %>%
  filter(is.na(RATE)==FALSE) %>%
  summarise(RATE_K200 = prod(1 + RATE),
            MMF = mean(MMF)/100,
            .groups = "drop") %>%
  ungroup()

# 3-1. 포트폴리오 구성 : VW양매도 포트폴리오 기초정보 생성
target_info <- target_date %>%
  left_join(options, by="BAS_DD") %>%
  distinct(BAS_DD,PRIOR) %>%
  arrange(desc(BAS_DD),PRIOR) %>%
  group_by(BAS_DD) %>%
  # 만기가 도래하는 옵션을 제외하고 우선순위가 높은 옵션 선택
  slice(2) %>%
  ungroup() %>%
  arrange(desc(BAS_DD)) %>%
  left_join(idx_data, by="BAS_DD") %>%
  left_join(mmf_raw, by="BAS_DD") %>%
  # 옵션 보유기간 및 단기금리(MMF) 계산
  mutate(DIFF_DAY=as.integer(ymd(lag(BAS_DD))-ymd(BAS_DD)),
         MMF=MMF*0.01) %>%
```

```
  # 옵션 보유기간에 해당하는 시장기대변동성 계산(V-K200활용)
  mutate(VOL=LAG_VK200*sqrt(DIFF_DAY)/sqrt(365)/100) %>%
  # 변동성에 따른 옵션 행사가격 상단(2.5단위 올림) 및 하단(2.5단위 내림) 계산
  mutate(TARGET_C_K=ceiling(KOSPI200*(1+VOL)*0.4)/0.4,
         TARGET_P_K=floor(KOSPI200*(1-VOL)*0.4)/0.4) %>%
  # drop_na(DIFF_DAY) %>%
  select(BAS_DD,VOL,DIFF_DAY,MMF,KOSPI200,LAG_VK200,PRIOR,TARGET_C_K,TARGET_P_K)

# 3-2. 포트폴리오 구성 : VW양매도 포트폴리오 거래대상 옵션 정보 생성
target_options <- target_info %>%
  select(BAS_DD, PRIOR, TARGET_C_K, TARGET_P_K) %>%
  pivot_longer(cols=c("TARGET_C_K","TARGET_P_K"),names_to = "GRP", values_to = "EXER_PRC") %>%
  mutate(RGHT=substr(GRP,8,8)) %>%
  left_join(options,by=c("BAS_DD","PRIOR","RGHT","EXER_PRC")) %>%
  select(BAS_DD,GRP,PRICE,ACC_TRDVOL) %>%
  pivot_wider(names_from = "GRP", values_from = c("PRICE","ACC_TRDVOL"))

# 원금 100억원 가정
cash = 10000*10000*100

# 4. 포트폴리오 구현 : VW양매도 전략을 구현하여 일자별 손익 계산
portfolio <- target_info %>%
  left_join(target_options,by="BAS_DD") %>%
  arrange(BAS_DD) %>%
  # 원금에 따른 옵션 매도수량 계산
  mutate(SELL_AMT=cash/250000/KOSPI200) %>%
  # 옵션 프리미엄(매도수익)) 계산
  mutate(PREMIUM=SELL_AMT*(PRICE_TARGET_C_K+PRICE_TARGET_P_K)*250000) %>%
  # 원금 및 프리미엄의 MMF 이자수익 및 권리행사손실 계산
  mutate(INTEREST=(cash+replace_na(PREMIUM,0))*MMF*DIFF_DAY/365,
         EXERCISE=(if_else(lead(KOSPI200)-TARGET_C_K>0,lead(KOSPI200)-TARGET_C_K,0)+
                     if_else(TARGET_P_K-lead(KOSPI200)>0,TARGET_P_K-lead(KOSPI200),0))*250000*
  # 주단위 포트폴리오 손익 계산(원금 100억 가정, 당일 투자시 다음주 실현손익을 의미)
  mutate(REVENUE=PREMIUM+INTEREST-EXERCISE) %>%
  # 거래대상 옵션이 상장되어있지 않은 경우, MMF수익만 고려
  mutate(REVENUE=if_else(is.na(REVENUE),INTEREST,REVENUE)) %>%
  mutate(RATE=REVENUE/cash,
         Group="Vol-adj. Weekly")  # 주단위 포트폴리오 수익률
```

```r
# 5. 비교군 포트폴리오 생성 : 월별 5% OTM 양매도 전략
options2 <- options %>%
  filter(PROD=="코스피200")

# 옵션 만기일(매달) 생성
target_date2 <- options2 %>%
  distinct(.,BAS_DD, PRIOR) %>%
  arrange(PRIOR, desc(BAS_DD)) %>%
  group_by(PRIOR) %>%
  slice(1) %>%
  ungroup() %>%
  distinct(BAS_DD) %>%
  arrange(desc(BAS_DD)) %>%
  filter(BAS_DD<20250411)

# 일반 양매도 포트폴리오 기본 정보 생성
target_info2 <- target_date2 %>%
  left_join(options2, by="BAS_DD") %>%
  distinct(BAS_DD,PRIOR) %>%
  arrange(desc(BAS_DD),PRIOR) %>%
  group_by(BAS_DD) %>%
  # 만기가 도래하는 옵션을 제외하고 우선순위가 높은 옵션 선택
  slice(2) %>%
  ungroup() %>%
  arrange(desc(BAS_DD)) %>%
  left_join(idx_data, by="BAS_DD") %>%
  left_join(mmf_raw, by="BAS_DD") %>%
  # 옵션 보유기간 및 단기금리(MMF) 계산
  mutate(DIFF_DAY=as.integer(ymd(lag(BAS_DD))-ymd(BAS_DD)),
         MMF=MMF*0.01,
         VOL=0.05) %>%
  # 변동성에 따른 옵션 행사가격 상단(2.5단위 올림) 및 하단(2.5단위 내림) 계산
  mutate(TARGET_C_K=ceiling(KOSPI200*(1+VOL)*0.4)/0.4,
         TARGET_P_K=floor(KOSPI200*(1-VOL)*0.4)/0.4) %>%
  mutate(DIFF_DAY=if_else(is.na(DIFF_DAY),28,DIFF_DAY)) %>%
  select(BAS_DD,VOL,DIFF_DAY,MMF,KOSPI200,LAG_VK200,PRIOR,TARGET_C_K,TARGET_P_K)

# 일반 양매도 포트폴리오 거래대상 옵션
target_options2 <- target_info2 %>%
```

```
  select(BAS_DD, PRIOR, TARGET_C_K, TARGET_P_K) %>%
  pivot_longer(cols=c("TARGET_C_K","TARGET_P_K"),names_to = "GRP", values_to = "EXER_PRC") %>%
  mutate(RGHT=substr(GRP,8,8)) %>%
  left_join(options2,by=c("BAS_DD","PRIOR","RGHT","EXER_PRC")) %>%
  select(BAS_DD,GRP,PRICE,ACC_TRDVOL) %>%
  pivot_wider(names_from = "GRP", values_from = c("PRICE","ACC_TRDVOL"))

# 일반 양매도 포트폴리오 구현
portfolio2 <- target_info2 %>%
  left_join(target_options2,by="BAS_DD") %>%
  arrange(BAS_DD) %>%
  # 원금에 따른 옵션 매도수량 계산
  mutate(SELL_AMT=cash/250000/KOSPI200) %>%
  # 옵션 프리미엄(매도수익)) 계산
  mutate(PREMIUM=SELL_AMT*(PRICE_TARGET_C_K+PRICE_TARGET_P_K)*250000) %>%
  # 원금 및 프리미엄의 MMF 이자수익 및 권리행사손실 계산
  mutate(INTEREST=(cash+replace_na(PREMIUM,0))*MMF*DIFF_DAY/365,
         EXERCISE=(if_else(lead(KOSPI200)-TARGET_C_K>0,lead(KOSPI200)-TARGET_C_K,0)+
                     if_else(TARGET_P_K-lead(KOSPI200)>0,TARGET_P_K-lead(KOSPI200),0))*250000*
  mutate(EXERCISE=if_else(is.na(EXERCISE),0,EXERCISE)) %>%
  # 주단위 포트폴리오 손익 계산(원금 100억 가정, 당일 투자시 다음주 실현손익을 의미)
  mutate(REVENUE=PREMIUM+INTEREST-EXERCISE) %>%
  # 거래대상 옵션이 상장되어있지 않은 경우, MMF수익만 고려
  mutate(REVENUE=if_else(is.na(REVENUE),INTEREST,REVENUE)) %>%
  mutate(RATE=REVENUE/cash,
         Group="Monthly") # 주단위 포트폴리오 수익률

portfolio <- portfolio %>% filter(BAS_DD!=20250410)
portfolio2 <- portfolio2 %>% filter(BAS_DD!=20250410)
portfolio3 <- portfolio %>%
  summarise(across(where(is.numeric), sum)) %>%
  mutate(Group="Vol-adj. Weekly Sum")


# 요약표 4 : 검증 자료
table4 <- portfolio %>%
  union_all(portfolio2) %>%
  union_all(portfolio3) %>%
  group_by(BAS_DD, Group) %>%
```

```
  summarise(Premium = round(mean(PREMIUM)/10000,0),
            Interest = round(mean(INTEREST)/10000,0),
            Loss = round(mean(EXERCISE)/10000,0),
            Revenue = round(mean(REVENUE)/10000,0),
            Return = round(prod(1 + RATE)-1,2),
            .groups = "drop")

# 그래프 4 : 검증기간중 코스피200지수 추이
graph4 <- ggplot(idx_raw, aes(x = ymd(BAS_DD))) +
  geom_line(aes(y = KOSPI200, color = "KOSPI200"), size = 1.2) +
  geom_line(aes(y = (VKOSPI200 - 19) / 26 * 60 + 300, color = "VKOSPI200"), size = 1.2) +
  geom_vline(xintercept = ymd(c("20250313", "20250320", "20250327", "20250403","20250410")),
             linetype = "dotted", color = "grey40") +
  scale_y_continuous(name = "KOSPI200",limits = c(300, 360),
                     sec.axis = sec_axis(
                       ~ (.-300) / 60 * 26 + 19, name = "VKOSPI200")) +
  scale_color_manual(values = c("KOSPI200" = "blue", "VKOSPI200" = "red"),
                     guide = guide_legend(direction = "horizontal")) +
  labs(title = "Kospi200 & V-Kospi200 indices",x = NULL,color = NULL) +
  theme_minimal() +
  scale_x_date(date_breaks = "3 days", date_labels = "%m-%d") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = c(0.5, 0.95))
```

Part IV

장외파생상품 기초 실무('25 봄)

장외파생상품 기초실무 과제

20249132 김형환

1. 평가 상품 개요

이번 과제에서 평가해볼 장외파생상품은 S&P500 기반의 ELB입니다.

Long put + 20% Knock-out 형태로, 원금보장형 payoff를 가지고 있습니다.

세부 사항은 아래 사진과 같습니다.

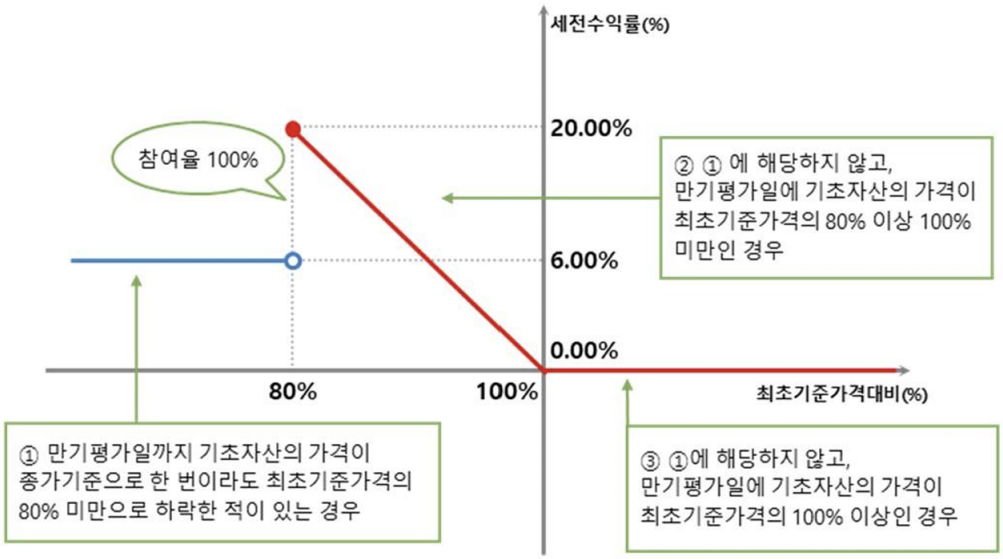
□ 상품개요

항 목		내용
종목명		KB able ELB 제214호 파생결합사채(주가연계파생결합사채) (낮은위험 / 5등급)
기초자산		S&P500 지수
모집총액		10,000,000,000원
최소청약금액		1,000,000원
청약기간	청약시작일	2025년 03월 12일
	청약종료일	2025년 03월 19일 (14:00까지) (청약기간 이후 청약취소 불가)
환불일		2025년 03월 19일 (발행 취소된 경우) 2025년 03월 19일 (안분 배정이 발생한 경우)
발행일		2025년 03월 19일
만기일		2026년 03월 19일
청약단위		최소청약금액 100만원으로 하되 100만원 이상은 10만원 단위로 합니다.

□ 권리의 내용

(1) 평가일, 관찰일 및 평가방법

○ **최초기준가격** : 최초기준가격평가일 기초자산 종가
○ **최초기준가격평가일** : 2025년 03월 19일
○ **만기평가가격** : 만기평가일 기초자산 종가
○ **만기평가일** : 2026년 03월 16일
○ **만기일** : 만기평가일(불포함) 후 **3 영업일**



이는 **배리어옵션**의 한 종류로, 만기까지 한번이라도 배리어(20%) 밑으로 하락하는 경우가 존재하면 옵션 권리가 사라지고(**Down-and-Out**), 대신 6%의 rebate를 지급하는 구조입니다.

2. 평가

평가 개요

평가는 몬테카를로 시뮬레이션을 이용할 계획이며, Numerix pricer와 동일한 2025-4-29일을 기준으로 pricing하도록 하겠습니다. 먼저, 배리어옵션 평가를 위한 parameter는 아래와 같이 정리하였습니다.

- S_0 : S&P500지수의 최초기준가격평가일('25.3.19) 종가 (5,675.29pt)
- S_1 : S&P500지수의 평가일('25.4.29) 종가 (5,560.83pt)
- K : Down-and-out put options 행사가격(=$S_0 \times 100\%$)
- B : Down-and-out put options 배리어가격(=$S_0 \times 80\%$)
- r : 무위험이자율(=3% 가정)
- q : S&P500 배당수익률(=0% 가정정)
- σ : 기초자산 변동성(=20% 가정)
- T_1 : 잔존만기(=324/365)
- T_2 : 배리어옵션 평가대상만기(=321/365)
- n : 몬테카를로 시뮬레이션 시행횟수
- m : 시뮬레이션 경로 분할 갯수(거래일별, 220)
- $rebate$: 배리어옵션 Knock-out 시 지급하는 쿠폰(=6%)

> **i** 파라미터 결정 근거
>
> 상품 평가를 위한 파라미터의 산정 근거는 아래와 같습니다.
>
> - 기초자산가격, 행사가격, 배리어가격, 만기 등 : 상품 개요에서 확인

- 무위험이자율 : 미국에서는 SOFR, USD-libor, FFR 등이 널리 사용되며, Swap rate 등을 활용해 term-structure를 구성하는 것이 가장 정밀한 방법입니다. 분석의 단순화를 위해 3%로 가정하였습니다.

- S&P500 배당수익률 : 과거 배당이 유지된다고 가정하거나, 별도의 모델링을 통해 기간구조를 구성하는 것이 가장 정밀한 방법입니다. 분석의 단순화를 위해 배당이 없다고 가정하였습니다.

- 평가대상만기 : 만기평가일에 옵션 payoff가 결정되므로, 3거래일을 차감한 값을 활용하였습니다.

- 기초자산 변동성 : 시장가격을 활용한 내재변동성을 이용하였으며, Local Vol 등을 통해 Surface를 구성하여 시뮬레이션 하는 것이 가장 정밀한 방법입니다. 다만, 분석의 단순화를 위해 20%로 가정하였습니다.

- 경로 분할 갯수 : Lookback period의 간격이 1일이므로 거래일수로 분할하였습니다.

거래일은 NYSE를 기준으로 산출하였고, 데이터는 yahoo finance, cme, cboe를 참고하였습니다.

```
import pandas_market_calendars as mcal
from datetime import date

exp_dd = date(2026, 3, 19); strt_dd = date(2025, 4, 29);
exptime = (exp_dd - strt_dd).days

strt = '2025-04-30'; end='2026-03-16'; nyse = mcal.get_calendar('NYSE');
timesteps = nyse.valid_days(start_date=strt, end_date=end)

print("Day to maturity is :",exptime)
print("Number of trading day(timestep) is :",len(timesteps))
```

```
Day to maturity is : 324
Number of trading day(timestep) is : 220
```

평가 알고리즘

기초자산인 **S&P500 지수가 GBM을 따른다는 가정** 하에 정규난수를 이용하여 만기평가일까지의 주가흐름을 시뮬레이션할 계획이며, 알고리즘은 아래와 같습니다.

(1) 현재 주가(S_0)와 만기(T_2), 변동성(σ), 기대수익률($\mu = r - d$)를 통해 GBM을 구성하고, Euler's discretization을 통해 248거래일마다 종가를 생성

- 먼저, 평가대상만기일의 종가를 n개 생성(Stratified sampling, Moment matching, Antithetic variate를 활용하여 균질한 분포를 구현)

- 각 n개의 종가마다, 그 경로의 247(총 (m-1)*n)개의 난수를 생성(Antithetic variate 적용)하고, brownian bridge 방법을 통해 오일러 이산화를 구현

(2) 각 경로마다 Knock 여부와 내재가치를 고려하여 배리어옵션의 payoff와 npv를 결정

- Knock-out 발생 : 명목금액의 6%의 rebate를 지급받음
- Knock-out 미발생 : 평가만기일 종가에 따라 풋옵션 내재가치를 지급받음
- 각 payoff를 잔존만기(T_1)에 대해 할인하여 npv를 산출

(3) 이에 따라 산출된 npv를 산술평균을 통해 배리어옵션(ELB)의 공정가치를 산출

알고리즘 구현 (Python code)

Python으로 구현하였으며, 배리어옵션의 가격과 기초자산의 경로(GBM)를 반환하는 함수로 작성하였습니다.

```
import numpy as np
import scipy.stats as sst

def DownAndOutPut_Price(s, k, r, q, t, sigma, n, b, m, rebate):
    dt = t/m; dts = np.arange(dt, t+dt, dt) # Set timesteps
    # (1) Stratified sampling, z_t makes price at T & z makes brownian bridge
    z_t = sst.norm.ppf((np.arange(n) + np.random.uniform(0,1,n)) / n)
    z = np.random.randn(n,m)
    # (2) Moment matching in z_t
    z_t = np.where(n>=100, (z_t - z_t.mean()) / z_t.std(ddof=1), z_t - z_t.mean())
    # (3) Antithetic variate
    z_t, z = np.concatenate([z_t, -z_t], axis=0), np.concatenate([z, -z], axis=0)
    # Generate underlying paths using brownian bridge
    w_t, w = z_t * np.sqrt(t), z.cumsum(axis=1) * np.sqrt(dt) # winner process
    bridge = dts * ((w_t- w[:,-1]).reshape(len(w),1) + w / dts) # brownian bridge
    paths = s*np.exp((r-q-0.5*sigma**2)*dts + sigma*bridge) # underlying path
    # Determine Knock-out
    knock = paths.min(axis=1) < b # knock-out = 1 else 0
    barrier_flag = ~knock
    # Caculate options payoff
    plain_npv = np.maximum(k-paths[:,-1], 0) * np.exp(-r*t)
    barrier_npv = barrier_flag * plain_npv + knock * rebate * np.exp(-r*t)
    barrier_price = barrier_npv.mean()

    return barrier_price, paths
```

3. 평가 결과 및 비교(Numerix Pricer)

배리어옵션 가격 및 기초자산 경로

상술한 2025-4-29일의 기준 파라미터를 이용하여 배리어옵션의 가격을 평가해보았습니다.

가격은 명목금액 **10,000원을 기준으로 약 365원이며, 수익률로 환산할 때 약 3.65%**입니다.

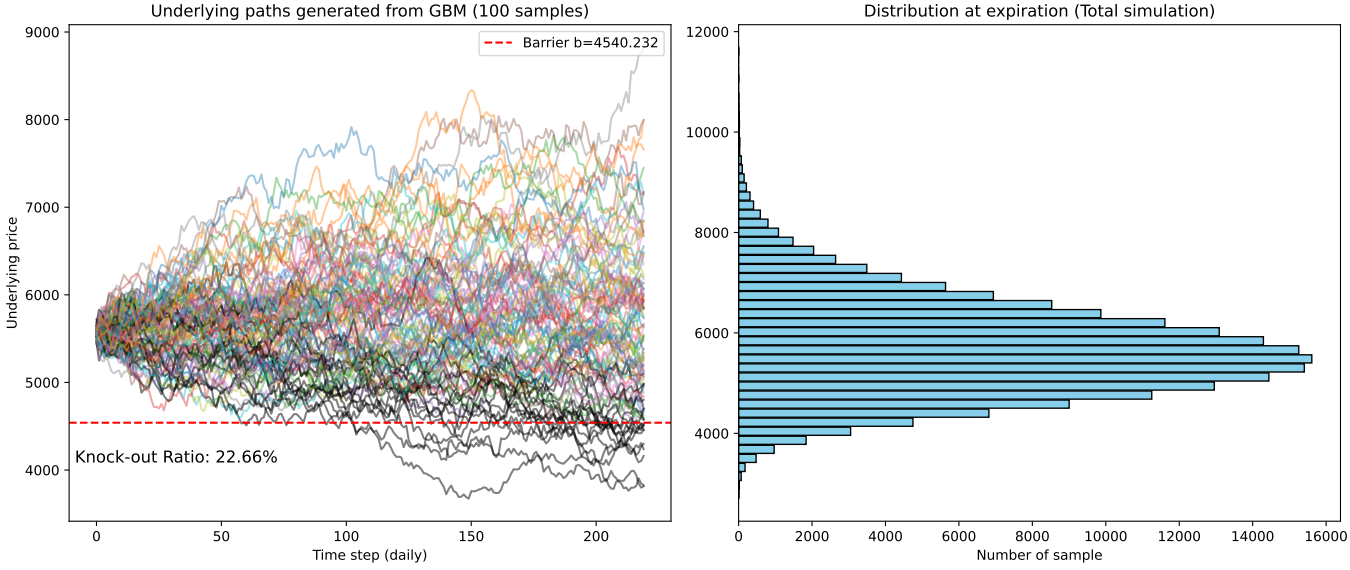
시뮬레이션 횟수는 10만번 기준으로, 난수에 따라 편차가 약 1~2원 존재하였습니다.

```
s0 = 5675.29; s1 = 5560.83; k = s0; b = 0.8 * s0
r = 0.03; q = 0; sigma = 0.2; rebate = 0.06 * s0
t1 = exptime / 365; t2 = (exptime-3) / 365; n = 100000; m = len(timesteps)
notional = 10000; nxprice = 366.8977875

price, GBMpath = DownAndOutPut_Price(s1, k, r, q, t2, sigma, n, b, m, rebate)
price = price * np.exp(r*(t2-t1)) / s0 * notional # Convert discount & notional
print(f"Down-and-Out Barrier Put options price is : {price:.4f}")
```

```
Down-and-Out Barrier Put options price is : 364.2851
```

시뮬레이션에 이용된 기초자산의 경로를 시각화한 결과입니다.

계층화, 표준화 등 분산감소기법이 잘 적용되어 **만기시점의 Lognormal dist. 형태**가 잘 나타난 것을 볼 수 있으며, 시뮬레이션 중 배리어가격 밑으로 하락한 경우가 발생한 **Knock-out의 비율은 약 22.7%**로 나타났습니다.



Numerix Pricer와 비교

파이썬에서 구현한 환경과 최대한 유사하도록 기존 Numerix pricer에서 무위험이자율 커브 및 변동성 곡면을 각각 3%, 20% 단일값으로 수정하였으며, 기초자산의 가격도 2025-4-29일 종가인 5560.83pt로 수정하였습니다.

이에 따라 수정된 Numerix를 이용한 배리어옵션 가격은 약 366.9원(3.67%)로 산출되었습니다.

EQ	
ID	MODEL.EQ.BS10715
OBJECT	MODEL
TYPE	EQ
MODEL	black
NOWDATE	2025-04-29
CURRENCY	USD
DIVIDENDCURVE	APR-2025-00-00-00^EQ.USD-US-SPX.DIV
DOMESTICYIELDCURVE	hwan_RFR
SPOTPRICE	5560.83
ASSETNAME	USDUSSPX
FIXCALENDAR	NewYork
FIXCONVENTION	F
NOTICEPERIOD	2BD
DISCRETEDIVIDENDCONV	
MARKETDATA	25-00-00-00^CONTAINER_QUOTES10110
INSTRUMENTS	QEUROPEANINSTRUMENTSCOMP_10993
SIGMA1	15.VOLATILITY.CURVECALIBRATION11431
LIBRARY NAME	TOOLKIT
OPTIONFILTER	15.OPTIONSFILTER.OPTIONSFILTER11436
CONTAINER	DD_SEL_20250429_29-APR-2025-00-00-00
UPDATED	5753 @ 09:22:22 PM
ID	APR-2025-00-00-00^MODEL.EQ.BS10715
LOCAL ID	MODEL.EQ.BS10715
TIMER	0.0323633
TIMER CPU	0.03125

Control Panel	
BarrierLevel	-0.2
StrikeLevel	0
CouponOut	0.06
Coupon	0
InitialPrice	5675.29
ParticipationRate	1
PV	366.8977875
CouponRate	0.037693897
HasKO	0.241

두 가격간의 괴리율은 1% 미만 수준으로, 상당히 유사한 것을 확인할 수 있었습니다.

Knock-out 비율도 Numerix에서 약 24%로 기존의 약 23%와 매우 유사하였습니다.

```
Difference ratio is : 0.0072
```

4. 소결

Down-and-out put options 형태의 ELB를 기본적인 Black-sholes 공식을 기반으로 **두가지 방식으로 평가**하였습니다.

1. Montecarlo Simulation : 기초자산의 가격경로를 생성하여 베리어옵션의 payoff를 평가
2. Numerix pricer : Numerix에서 제공하는 Black 모델 및 Kernel object를 이용하여 가격 평가

그 결과, **두 방식에서 매우 유사한 결과**를 얻을 수 있었으며 Knock-out 비율의 차이도 미미하여 기초자산의 가격경로의 생성방법이 비슷하다는 것을 유추할 수 있었습니다.

Numerix pricer에서는 기초적인 MCS방법 뿐만아니라, 이자율/배당/변동성 등 **기간구조를 가지는 변수**들을 object로 형성하고 이를 **쉽게 가격평가에 적용**할 수 있는 장점이 있었습니다. 그러나, **파이썬 환경**에서 이를 적용하기 위해서는 모든 난수생성 및 경로생성에 개입하여 Stocastic한 이자율/변동성 등 변수들을 일일히 적용해야하므로, **계산자원이 급격히 증가할 것으로 예상**됩니다.

본 분석의 **한계점**은 **다양한 변수조건에서의 검증이 생략**되어있다는 점 입니다. 상술한 기간구조를 반영하지 못한 부분뿐만아니라, 분석에서 다루지는 않았으나 **시장변동성이 커지는 상황**에서 MCS와 Numerix pricer의 **괴리율이 증가하는 경향**을 보였습니다.

부가적으로, QuantLib library를 이용하여 동일한 배리어옵션의 **Analytic price를 산출**해본 결과, **MCS 및 Numerix 방식과는 큰 차이**가 있었습니다. 이는 Analytic price를 산출할때 **연속시간을 가정**하므로, 실제 종가로만 판단하는 daily 시간간격대비 **Knock-out 비율이 높아져 옵션 가격을 저평가**하는 것으로 추정됩니다.

QuantLib을 이용한 배리어옵션 평가

```
import QuantLib as ql

today = ql.Date(29, 4, 2025); maturity = ql.Date(16, 3, 2026)
ql.Settings.instance().evaluationDate = today

payoff = ql.PlainVanillaPayoff(ql.Option.Put, k)
euExercise = ql.EuropeanExercise(maturity)
barrierOption = ql.BarrierOption(ql.Barrier.DownOut, b, rebate, payoff, euExercise)

spotHandle = ql.QuoteHandle(ql.SimpleQuote(s1))
flatRateTs = ql.YieldTermStructureHandle(ql.FlatForward(today, r, ql.Actual365Fixed()))
flatVolTs = ql.BlackVolTermStructureHandle(ql.BlackConstantVol(today, ql.NullCalendar(), sig
bsm = ql.BlackScholesProcess(spotHandle, flatRateTs, flatVolTs)

analyticBarrierEngine = ql.AnalyticBarrierEngine(bsm)
barrierOption.setPricingEngine(analyticBarrierEngine)
analytic_price = barrierOption.NPV() * np.exp(r*(t2-t1)) / s0 * notional

print(f"Analytic Price is : {analytic_price:.4f}")
```

```
Analytic Price is : 357.1444
```

Part V

금융윤리와 사회책임('25 봄)

금융윤리와 사회책임

20249132 김형환

Standard 1 : Professionalism

A. 법규의 이해와 준수 (Knowledge of the Law)

A. Knowledge of the Law

Members and Candidates must understand and comply with all applicable laws, rules, and regulations (including the CFA Institute Code of Ethics and Standards of Professional Conduct) of any government, regulatory organization, licensing agency, or professional association governing their professional activities. In the event of conflict, Members and Candidates must comply with the most strict law, rule, or regulation. Members and Candidates must not knowingly participate or assist in and must dissociate from any violation of such laws, rules, or regulations.

법규의 이해와 준수

회원과 응시생들은 자신들의 전문적 행위를 지배하는 정부, 규제기관, 면허발급기관 혹은 전문가협회가 정한 관련된 법률, 규칙, 규정(CFA협회의 윤리강령 및 행위기준을 포함하여)을 이해하고 이를 준수해야 한다. 상충되는 경우에는 보다 엄격한(more strict) 법률, 규칙 및 규정을 따라야 한다. 해당 법률, 규칙, 규정에 대한 위반을 알면서도 가담하거나 도와서는 안되며, 일체의 위반행위와의 관계를 단절해야 한다.

- Relationship between the Code and Standards and Applicable Law
- Participation in or Association with Violations by Others
- Investment Products and Applicable Laws

2. Following the Highest Requirements

Laura Jameson works for a multinational investment advisor based in the United States. Jameson lives and works as a registered investment advisor in the tiny, but wealthy, island nation of Karramba. Karramba's securities laws state that no investment advisor registered and working in that country can participate in initial public offerings (IPOs) for the advisor's personal account. Jameson, believing that as a U.S. citizen working for a U.S.-based company she need comply only with U.S. law, has ignored this Karrambian law. In addition, Jameson believes that, as a charterholder, as long as she adheres to the Code and Standards requirement that she disclose her participation in any IPO to her employer and clients when such ownership creates a conflict of interest, she is meeting the highest ethical requirements.

Comment: Jameson is in violation of Standard I(A). As a registered investment advisor in Karramba, Jameson is prevented by Karrambian securities law from participating in IPOs regardless of the law of her home country. In addition, because the law of the country where she is working is stricter than the Code and Standards, she must follow the stricter requirements of the local law rather than the requirements of the Code and Standards.

4. Failure to Maintain Knowledge of the Law

Colleen White is excited to use new technology to communicate with clients and potential clients. She recently began posting investment information, including performance reports and investment opinions and recommendations, to her Facebook page. In addition, she sends out brief announcements, opinions, and thoughts via her Twitter account (for example, "Prospects for future growth of XYZ company look good! #makingmoney4U"). Prior to White's use of these social media platforms, the local regulator had issued new requirements and guidance governing online electronic communication. White's communications appear to conflict with the recent regulatory announcements.

Comment : White is in violation of Standard I(A) because her communications do not comply with the existing guidance and regulation governing use of social media. White must be aware of the evolving legal requirements pertaining to new and dynamic areas of the financial services industry that are applicable to her. She should seek guidance from appropriate, knowledgeable, and reliable sources, such as her firm's compliance department, external service providers, or outside counsel, unless she diligently follows legal and regulatory trends affecting her professional responsibilities.

B. 독립성과 객관성 (Independence and Objectivity)

B. Independence and Objectivity

Members and Candidates must use reasonable care and judgment to achieve and maintain independence and objectivity in their professional activities. Members and Candidates must not offer, solicit, or accept any gift, benefit, compensation, or consideration that reasonably could be expected to compromise their own or another's independence and objectivity.

독립성과 객관성

회원과 응시생들은 전문가 행위에 있어서 독립성과 객관성을 성취하고 유지하기 위해 합리적 주의와 판단을 다해야 한다. 자신 혹은 다른 사람의 독립성과 객관성을 위태롭게 할 것이라고 당연히 예상되는 어떠한 선물, 편익, 보상 혹은 보답 등을 제안하거나 간청하거나 수용해서는 안 된다

- Buy-Side Clients
- Fund Manager and Custodial Relationships
- Investment Banking Relationships
- Performance Measurement and Attribution
- Public Companies
- Credit Rating Agency Opinions
- Influence during the Manager Selection/Procurement Process
- Issuer-Paid Research
- Travel Funding

5. Influencing Manager Selection Decisions

Adrian Mandel, CFA, is a senior portfolio manager for ZZYY Capital Management who oversees a team of investment professionals who manage labor union pension funds. A few years ago, ZZYY sought to win a competitive asset manager search to manage a significant allocation of the pension fund of the United Doughnut and Pretzel Bakers Union (UDPBU). UDPBU's investment board is chaired by a recognized key decision maker and long-time leader of the union, Ernesto Gomez. To improve ZZYY's chances of winning the competition, Mandel made significant monetary contributions to Gomez's union reelection campaign fund. Even after ZZYY was hired as a primary manager of the pension, Mandel believed that his firm's position was not secure. Mandel continued to contribute to Gomez's reelection campaign chest as well as to entertain lavishly the union leader and his family at top restaurants on a regular basis. All of Mandel's outlays were routinely handled as marketing expenses reimbursed by ZZYY's expense accounts and were disclosed to his senior management as being instrumental in maintaining a strong close relationship with an important client.

Comment : Mandel not only offered but actually gave monetary gifts, benefits, and other considerations that reasonably could be expected to compromise Gomez's objectivity. Therefore, Mandel was in violation of Standard I(B).

C. 오해의 소지가 있는 표현 금지 (Misrepresentation)

C. Misrepresentation

Members and candidates must not knowingly make any misrepresentations relating to investment analysis, recommendations, actions, or other professional activities.

오해의 소지가 있는 표현(허위진술)금지

회원과 응시생들은 투자분석, 투자추천, 투자행위 혹은 기타 전문가행위와 관련하여 고의로 (knowingly) 오해의 소지가 있는 표현(허위진술)을 해서는 안 된다.

- Impact on Investment Practice
- Performance Reporting
- Social Media
- Omissions
- Plagiarism
- Work Completed for Employer

4. Plagiarism

Claude Browning, a quantitative analyst for Double Alpha, Inc., returns in great excitement from a seminar. In that seminar, Jack Jorrely, a well publicized quantitative analyst at a national brokerage firm, discussed one of his new models in great detail, and Browning is intrigued by the new concepts. He proceeds to test this model, making some minor mechanical changes but retaining the concept, until he produces some very positive results. Browning quickly announces to his supervisors at Double Alpha that he has discovered a new model and that clients and prospective clients alike should be informed of this positive finding as ongoing proof of Double Alpha's continuing innovation and ability to add value.

Comment: Although Browning tested Jorrely's model on his own and even slightly modified it, he must still acknowledge the original source of the idea. Browning can certainly take credit for the final, practical results; he can also support his conclusions with his own test. The credit for the innovative thinking, however, must be awarded to Jorrely.

6. Avoiding a Misrepresentation

Trina Smith is a fixed-income portfolio manager at a pension fund. She has observed that the market for highly structured mortgages is the focus of salespeople she meets and that these products represent a significant number of trading opportunities. In discussions about this topic with her team, Smith learns that calculating yields on changing cash flows within the deal structure requires very specialized vendor software. After more research, they find out that each deal is unique and that deals can have more than a dozen layers and changing cash flow priorities. Smith comes to the conclusion that, because of the complexity of these securities, the team cannot effectively distinguish between potentially good and bad investment options. To avoid misrepresenting their understanding, the team decides that the highly structured mortgage segment of the securitized market should not become part of the core of the fund's portfolio ; they will allow some of the less complex securities to be part of the core.

Comment : Smith is in compliance with Standard 1(C) by not investing in securities that she and her team cannot effectively understand. Because she is not able to describe the risk and return profile of the securities to the pension fund beneficiaries and trustees, she appropriately limits the fund's exposure to this sector.

Standard 2 : Integrity of Capital Markets

A. 중요한 미공개정보 (Material Nonpublic Information)

A. Material Nonpublic Information

Members and Candidates who possess material nonpublic information that could affect the value of an investment must not act or cause others to act on the information.

중요한 미공개 정보

투자대상의 가치에 영향을 미칠 수 있는 중요한 미공개정보를 소지한 회원과 응시생들은 이를 활용하여 투자행위를 하거나 다른 사람에게 행위를 하도록 해서는 안 된다.

- What Is "Material" Information?
- What Constitutes "Nonpublic" Information?
- Mosaic Theory
- Social Media
- Using Industry Experts
- Investment Research Reports

4. Materiality Determination

Larry Nadler, a trader for a mutual fund, gets a text message from another firm's trader, whom he has known for years. The message indicates a software company is going to report strong earnings when the firm publicly announces in two days. Nadler has a buy order from a portfolio manager within his firm to purchase several hundred thousand shares of the stock. Nadler is aggressive in placing the portfolio manager's order and completes the purchases by the following morning, a day ahead of the firm's planned earnings announcement.

Comment: There are often rumors and whisper numbers before a release of any kind. The text message from the other trader would most likely be considered market noise. Unless Nadler knew that the trader had an ongoing business relationship with the public firm, he had no reason to suspect he was receiving material nonpublic information that would prevent him from completing the trading request of the portfolio manager.

B. 시장조작 (Market Manipulation)

B. Market Manipulation

Members and Candidates must not engage in practices that distort prices or artificially inflate trading volume with the intent to mislead market participants.

시장조작

회원과 응시생들은 시장참여자들을 誤導(오도)할 목적으로 가격을 왜곡하거나 인위적으로 거래량을 부풀리는 행위에 가담해서는 안 된다.

- Information-Based Manipulation
- Transaction-Based Manipulation

3. Personal Trading and Volume

Rajesh Sekar manages two funds—an equity fund and a balanced fund—whose equity components are supposed to be managed following the same model. According to that model, the funds' holdings in stock of Digital Design Inc. (DD) are excessive. Reduction of the DD holdings would not be easy because the stock has low liquidity in the stock market. Sekar decides to start trading larger portions of DD stock back and forth between his two funds to slowly increase the price, believing that market participants would see growing volume and increasing price and become interested in the stock. If other investors are willing to buy the DD stock because of such interest, then Sekar would be able to get rid of at least some part of his overweight position without inducing price decreases. In this way, the whole transaction will be for the benefit of fund participants, even if additional brokers' commissions are incurred.

Comment: Sekar's plan would be beneficial for his funds' participants but is based on artificial distortion of both trading volume and price of DD stock and thus constitutes a violation of Standard II(B).

5. Manipulating Model Inputs

Bill Mandeville supervises a structured financing team for Superior Investment Bank. His responsibilities include packaging new structured investment products and managing Superior's relationship with relevant rating agencies. To achieve the best rating possible, Mandeville uses mostly positive scenarios as model inputs-scenarios that reflect minimal downside risk in the assets underlying the structured products. The resulting output statistics in the rating request and underwriting prospectus support the idea that the new structured products have minimal potential downside risk. Additionally, Mandeville's compensation from Superior is partially based on both the level of the rating assigned and the successful sale of new structured investment products but does not have a link to the long-term performance of the instruments. Mandeville is extremely successful and leads Superior as the top originator of structured investment products for the next two years. In the third year, the economy experiences difficulties and the values of the assets underlying structured products significantly decline. The subsequent defaults lead to major turmoil in the capital markets, the demise of Superior Investment Bank, and Mandeville's loss of employment.

Comment : Mandeville manipulates the inputs of a model to minimize associated risk to achieve higher ratings. His understanding of structured products allows him to skillfully decide which inputs to include in support of the desired rating and price. This information manipulation for short-term gain, which is in violation of Standard II(B), ultimately causes significant damage to many parties and the capital markets as a whole. Mandeville should have realized that promoting a rating and price with inaccurate information could cause not only a loss of price confidence in the particular structured product but also a loss of investor trust in the system. Such loss of confidence affects the ability of the capital markets to operate efficiently.

Standard 3 : Duties to Clients

A. 고객에 대한 충성의무 (Loyalty, Prudence, and Care)

A. Loyalty, Prudence, and Care

Members and Candidates have a duty of loyalty to their clients and must act with reasonable care and exercise prudent judgment. Members and candidates must act for the benefit of their clients and place their clients' interests before their employer's or their own interests.

고객에 대한 충성의무

회원과 응시생은 고객에 대한 충성의무가 있으며 합리적으로 행동하고, 신중한 판단을 해야 한다. 회원과 응시생은 고객의 이익을 위해 행동해야 하며, 고객의 이익을 자신이나 회사의 이익에 우선해야 한다.

- Understanding the Application of Loyalty, Prudence, and Care
- Identifying the Actual Investment Client
- Developing the Client's Portfolio
- Soft Commission Policies
- Proxy Voting Policies

3. Managing Family Accounts

Adam Dill recently joined New Investments Asset Managers. To assist Dill in building a book of clients, both his father and brother opened new fee-paying accounts. Dill followed all the firm's procedures in noting his relationships with these clients and the developing their investment policy statements.

After several years, the number of Dill's clients has grown, but he still manages the original accounts of his family members. An IPO is coming to market that is a suitable investment for many of his clients, including his brother. Dill does not receive the amount of stock he requested, so to avoid any appearance of a conflict of interest, he does not allocate any shares to his brother's account.

Comment : Dill has violated Standard III(A) because he is not acting for the benefit of his brother's account as well as his other accounts. The brother's account is a regular fee-paying account comparable to the accounts of his other clients. By not allocating the shares proportionately across all accounts for which he thought the IPO was suitable, Dill is disadvantaging specific clients.
Dill would have been correct in not allocating shares to his brother's account if that account was being managed outside the normal fee structure of the firm.

B. 차별금지 (Fair Dealing)

B. Fair Dealing

Members and Candidates must deal fairly and objectively with all clients when providing investment analysis, making investment recommendations, taking investment action, or engaging in other professional activities.

차별금지

회원과 응시생들은 투자분석의 제공, 투자추천, 투자행위, 기타 전문가적 행위를 수행함에 있어서 모든 고객을 공정하고 객관적으로 대해야 한다.

- Investment Recommendations
- Investment Action

2. Fair Dealing and Transaction Allocation

Eleanor Preston, the chief investment officer of Porter Williams Investments (PWI), a medium-sized money management firm, has been trying to retain a difficult client, Colby Company. Management at the disgruntled client, which accounts for almost half of PWI's revenues, recently told Preston that if the performance of its account did not improve, it would find a new money manager. Shortly after this threat, Preston purchases mortgage-backed securities (MBS) for several accounts, including Colby's. Preston is busy with a number of transactions that day, so she fails to allocate the trades immediately or write up the trade tickets. A few days later, when Preston is allocating trades, she notes that some of the MBS have significantly increased in price and some have dropped. Preston decides to allocate the profitable trades to Colby and spread the losing trades among several other PWI accounts.

Comment: Preston violated Standard III(B) by failing to deal fairly with her clients in taking these investment actions. Preston should have allocated the trades prior to executing the orders, or she should have had a systematic approach to allocating the trades, such as pro rata, as soon after they were executed as practicable.
Among other things, Preston must disclose to the client that the advisor may act as broker for, receive commissions from, and have a potential conflict of interest regarding both parties in agency cross transactions. After the disclosure, she should obtain from the client consent authorizing such transactions in advance.

C. 투자의 적합성 (Suitability)

C. Suitability

1. When members and candidates are in an advisory relationship with a client, they must :

a. Make a reasonable inquiry into a client or prospective client's investment experience, risk and return objectives, and financial constraints prior to making any investment recommendation or taking investment action and must reassess and update this information regularly.

b. Determine that an investment is suitable to the client's financial situation and consistent with the client's written objectives, mandates, and constraints prior to making an investment recommendation or taking investment action.

c. Judge the suitability of investments in the context of the client's total portfolio.

2. When members and candidates are responsible for managing a portfolio to a specific mandate, strategy, or style, they must only make investment recommendations or take investment actions that are consistent with the stated objectives and constraints of the portfolio.

투자의 적합성

1. 회원과 응시생들은 고객과의 자문관계에서 다음과 같이 행동해야 한다 :

a. 투자추천 및 투자행위를 하기 전에 고객 혹은 잠재고객의 투자경험, 위험과 수익률목적, 그리고 재무적 제약조건 등을 합리적으로 조사, 파악하고 이러한 정보를 정규적으로 재평가하여 반영해야 한다.

b. 투자추천 및 투자행위를 하기 전에 그 투자대상이 고객의 재정상황에 적절한지 고객의 서면상 목적, 지시사항 및 제약조건에 부합되는지를 결정해야 한다.

c. 고객의 전체포트폴리오관점에서 투자의 적합성여부를 판단해야 한다.

2. 회원과 응시생들이 특정한 지시사항, 전략, 혹은 투자스타일에 따라 포트폴리오를 운용하는 책임을 맡고 있다면 그 포트폴리오에 명시된 목적과 제약조건에 부합되는 투자추천과 투자행위만을 수행해야 한다.

- Developing an Investment Policy
- Understanding the Client's Risk Profile
- Updating an Investment Policy
- The Need for Diversification
- Addressing Unsolicited Trading Requests
- Managing to an Index or Mandate

4. Investment Suitability-Risk Profile

Samantha Snead, a portfolio manager for Thomas Investment Counsel, Inc., specializes in managing public retirement funds and defined-benefit pension plan accounts, all of which have long-term investment objectives. A year ago, Snead's employer, in an attempt to motivate and retain key investment professionals, introduced a bonus compensation system that rewards portfolio managers on the basis of quarterly performance relative to their peers and to certain benchmark indices. In an attempt to improve the short-term performance of her accounts, Snead changes her investment strategy and purchases several high-beta stocks for client portfolios. These purchases are seemingly contrary to the clients' investment policy statements. Following their purchase, an officer of Griffin Corporation, one of Snead's pension fund clients, asks why Griffin Corporation's portfolio seems to be dominated by high-beta stocks of companies that often appear among the most actively traded issues. No change in objective or strategy has been recommended by Snead during the year.

Comment : Snead violated Standard III(C) by investing the clients' assets in high-beta stocks. These high-risk investments are contrary to the long-term risk profile established in the clients' IPS. Snead has changed the investment strategy of the clients in an attempt to reap short-term rewards offered by her firm's new compensation arrangement, not in response to changes in clients' investment policy statements.
See also standard VI(A)-Disclosure of Conflicts.

Assignment 1

기준1A : 법규의 이해와 준수

- CFA는 업무를 행하는데 있어서 관련 정부기관, 규제기관, 협회 등의 다양한 법규를 준수해야함
- 법규는 법률, 규칙, 규정, 윤리강령 및 행위기준 등 업무와 연관된 **일체의 규제 또는 가이드라인**을 포괄
- 이러한 법규의 위반을 알면서도 가담하거나 도와서는 안되며, 위반행위와 연관되어서는 아니됨
- **법규간 상충이 있는 경우 상위 법규**에 따라야 함

사례 2 : 법규간 상충시, 상위 법규를 따라야하나 이를 위반

- A는 미국계 다국적 투자자문사에서 일하면서, 개인 계좌로 공모주 거래를 희망
- A는 미국시민이고, 현재 카람바 소재의 지사에서 근무 및 거주
- 투자자문사 직원이 공모주 거래를 하는 경우, 미국은 합법 / 카람바는 불법
- A는 본인은 미국 시민이며, CFA 윤리강령에 따라 고객에게 공모주 거래사실을 공개하면 무관한 것으로 판단하여 공모주

> 시민으로서 미국법과 임직원으로서의 카람바법, 윤리강령이 상충하는 경우로 보임. 이 경우 **임직원으로서의 카람바 증권법이 상위법규**로서 준수해야하나 이를 무시하였으므로 A는 기준1A를 위반

사례 4 : 새로운 법규 도입 사실을 알지 못해 이를 위반

- B는 고객 소통 및 마케팅을 위해 소셜미디어 플랫폼을 적극 활용하고자 함
- 페이스북을 통해 성과보고서, 투자의견 및 추천정보 게재 및 트위터도 활용 예정
- 그러나, 해당국의 규제기관은 최근 소셜미디어 사용에 대한 새로운 가이드라인 발표
- B의 활동은 이 가이드라인과 상충되는 것으로 판단됨

> 비교적 최근에 규제가 발표되었고, **B가 이를 인지하지 못했다고 하더라도 관련 법규를 사전에 숙지**하고 준수하지 못하였으므로 B는 기준1A를 위반

Assignment 2

기준1C : 오해 소지가 있는 표현 금지

- CFA는 투자추천 등 전문행위와 관련하여 고의로 오해 소지가 있는 표현을 해서는 안됨

사례 4 : 본인이 원작자인 것처럼 표현하여 기준을 위반(표절)

- A는 퀀트애널리스트로, 세미나에서 B가 발표한 새로운 모델에 관심을 가지고 있음
- A는 새로운 모델을 기반으로 몇가지 수정을 거쳐서 자사에 맞는 모델로 최적화 성공
- 최적화 모델을 본인이 개발한 것처럼 책임자에게 보고하였고, 사업화 추진 제안

> 본인이 개발한 모델인 것처럼 모호한 표현을 사용하였으므로, A는 기준1C를 위반. 그러나, A가 모델을 최적화한 것은 공로이므로, **원작자가 B임을 명확하게 밝혔다면 최적화된 모델을 사업화하는 것은 기준 위반이 아님.**

사례 *6* : 오해의 소지를 피하고자 매우 복잡한 구조의 상품은 투자를 포기

- B는 연금펀드를 운용하는 채권매니저로, 최근 모기지 시장에 관심을 가지고 있음
- 구조화 모기지 증권에 투자하고자 하며, 다양한 기회가 존재함을 긍정적으로 평가
- 전문적인 소프트웨어로 평가해본 결과, 매우 복잡한 증권구조를 가지는 것을 확인
- 이로 인해, 각 증권마다 팀 차원에서 적절한 투자판단을 내리기 어렵다고 판단
- 따라서, 오해를 피하고자 덜 복잡한 모기지 증권에만 투자하는 것으로 결정

> B가 복잡한 구조화 모기지 증권에 투자하기로 하였다면, 매 증권마다 투자 판단을 내리기 위해 증권구조를 설명하고 이해하는 과정이 필요함. 이 과정에서 불필요한 오해가 발생할 수 있으므로, B는 기준1C를 준수하기 위해 이러한 오해를 최소화 한 것임

Assignment 3 : 1B, 2A, 2B

기준 1B : 독립성과 객관성 유지

- CFA는 직무수행에 있어서 독립성과 객관성을 유지하기 위해 합리적 주의와 판단을 다해야 함

사례 *5* : 고객 접대를 통해 독립성과 객관성 훼손

- A는 Z사의 연금기금을 운용하는 펀드매니저로, U사의 연금기금 운용사로 선정되기위해 노력중
- U사의 업무담당자 B에게 자금 기부 등 금전적 지원과 가족을 동행한 식사 접대 등 혜택을 제공함
- 이러한 지출은 모두 Z사의 마케팅 비용으로 처리되었으며, 커뮤니케이션 용도로 보고하였음

> 연금기금 운용사 선정 담당자인 B에게 직접적인 금전적 지원을 하는 등 업무수행에 있어서 독립성과 객관성을 훼손하였으므로 A는 기준1B를 위반

기준 2A : 미공개정보 이용 금지

- CFA는 미공개정보를 활용하여 투자행위를 하거나 다른 사람에게 하도록 해서는 안 됨

사례 *4* : 시장에 떠도는 루머를 이용하는 경우

- A는 운용사의 거래집행 트레이더로, 다른 회사의 B로 부터 Z사의 호실적이 예상된다는 루머를 받음
- 마침, A는 자사 포트폴리오 매니저로부터 Z사 주식의 매수 주문을 받았고, 빠르게 실행하였음

> 이러한 소문은 시장에 흔히 떠돌며, **직접적인 내부자로부터 받은 정보가 아니므로 미공개정보라고 보기 어려움.** 더군다나 A는 거래집행 트레이더로, **투자결정은 포트폴리오 매니저로부터 이루어**지고 단순히 거래집행만 하므로, A는 기준2A를 준수

기준 2B : 시장조작행위 금지

- CFA는 시장참여자를 오도할 목적으로 가격을 왜곡하거나 인위적으로 거래를 부풀리는 등 일체의 시장조작 행위를 해서는 안 됨

사례 *3* : 자전거래를 통한 거래량 왜곡

- A는 펀드매니저로, 주식형 펀드와 혼합형 펀드를 운용하고 있으며 동일하게 D사의 주식을 보유
- D사 주식을 두 펀드에서 모두 매도하고자 하나, 유동성이 부족하여 매도가 어려운 상황
- 두 펀드간 교차거래를 통해 인위적으로 거래량을 부풀렸고, 그 결과 투자자 관심을 끌어 원하는 수량만큼 매도를 희망

> 이는 **자전거래를 통해 거래량을 인위적으로 부풀려 왜곡하는 시장조작**행위에 해당하므로, A는 기준2B를 위반

사례 *5* : 모델 파라미터 조작을 통해 성과 평가 왜곡

- A는 투자은행의 구조화상품 설계 및 신용평가 담당자로, 그의 성과는 판매실적과 상품 신용등급에 연동
- 손실과는 연동되어있지 않아, 모델 파라미터를 조작하여 하방위험이 거의 없는 것처럼 보이게 하였음
- 이를 통해, 성공적인 신용등급과 판매실적을 이끌어냈으며 높은 성과급과 승진을 보상받았음
- 그러나, 시장 상황이 악화되면서 큰 손실이 발생하였고, 자본시장에 연쇄적인 충격을 야기

본인의 이익을 위해 모델 파라미터를 조작하였고, 회사와 시장 전체에 큰 손실을 미쳤으므로 중대한 위반에 해당함. A는 기준2B를 위반하였을 뿐만 아니라, 임직원으로서 기망, 사기행위이며 법률 위반의 소지가 있음.

Assignment 4 : 3A, 3B, 3C

기준 3A : 고객에 대한 충성의무

- CFA는 고객을 응대할 때, 합리적으로 행동하고 신중한 판단을 해야 함. 고객의 이익을 위해 행동해야하며 자신이나 회사의 이익에 우선해야 함.

사례 *3* : 가족계좌라 하더라도 고객의 이익을 우선해야 함

- A는 투자회사에 입사하였으며, 고객기반확보를 위해 가족이 투자계좌를 개설하고 A가 운용
- 가족계좌를 포함하여 A가 관리하는 유사한 계좌에 유리한 IPO정보를 입수함
- 그러나, 가족계좌를 우선하여 관리했다는 이해상충을 피하기 위해 모든 관리계좌에 IPO 미참여

> 가족계좌의 이해상충을 위해서라고 할지라도, **가족을 포함해서 유사한 계좌주들은 모두 고객에 해당**하므로 고객의 이익을 우선해야하는 충성의무 위반에 해당함. 즉, A는 기준3A를 위반하였음.

기준 3B : 차별금지

- CFA는 업무행위를 수행함에 있어 모든 고객을 공정하고 객관적으로 대해야 함.

사례 *2* : 특정 계좌에 이익이 나도록 분배하여 고객을 차별

- A는 자산운용사의 투자 책임자로, 최근 우량고객 C사가 성과 부진을 이유로 이탈 우려
- A는 성과를 위해 MBS를 과도하게 매수하였으며, 거래가 너무 많아 당일 고객계좌로 분배하지 못함.
- 몇일 뒤, MBS에서 수익이 많이 발생하자 C사의 계좌로 모두 분배하고, 손실은 타계좌로 분배하였음.

> 기존 투자결정 당시의 분배원칙에 맞게 고객계좌로 분배해야하나, C사의 이탈을 막고자 이익을 집중시켜 고객을 차별하였으므로 A는 기준3B를 위반하였음.

기준 3C : 투자의 적합성 판단

- CFA는 고객에게 자문할 때 고객의 투자경험, 위험과 목적, 재무조건 등을 합리적으로 조사, 파악해야 함. 또한 투자대상이 고객의 재정상황에 적절한지 고객에게 부합하는지 결정해야 함. 이는 전체 포트폴리오 관점에서 이루어져야 함.
- 더불어, 포트폴리오에 명시된 목적과 제약조건에 부합하는 자문만 수행하여야 함.

사례 *4* : 고객의 투자 정책을 무시하고 투자 집행

- A는 투자회사의 포트폴리오 매니저로, 은퇴기금이나 연금계좌 등 장기적인 자산을 관리함.
- 회사의 성과목표를 달성하기 위해, A의 독단적인 결정으로 고베타의 주식을 편입.
- 이는 고객의 투자 정책과 상반된 행동으로, 이러한 결정을 고객에게 보고하지 않았음.

장기적인 자산을 관리하는 고객의 투자 정책과 상반되었으므로, **A는 고객의 투자 적합성을 제대로 파악하고 판단하지 않았음**. 이는 A는 기준3C를 위반하였을 뿐만 아니라, 독단적인 결정으로 회사 평판 등에 손실을 입힐 수 있으므로 다른 문제도 야기할 수 있음.

Assignment 5

Standard 3-E : 고객기밀보호

사례 3 : SNS로 인한 우발적 기밀 누설

- A는 투자회사의 투자 담당 임원으로, 12명의 고액 자산가의 과세를 관리
- 고객들은 SNS를 통해 금융 및 경제 정보를 전달받고자 하며, A는 별도 페이지를 만들어 관리
- 그러나, SNS 특성상 보안에 취약할 수 있어 이에 대한 유의사항을 고객에게 안내
- 향후 고객 중 한명이 해당 페이지의 댓글에 개인 중요사항을 남겨 모든 고객이 이를 열람

A는 이러한 위험에 대해 고객들에게 사전에 충분히 안내하였으므로, 기준3E를 위반하지 않았음. 그러나 이러한 보안 취약점에 대해 향후 교육 등으로 보완해야할 것으로 보임.

Standard 4-A : 충성의무

사례 4 : 퇴사 후 현 직장의 사업 수주 가능여부

- A 및 몇몇의 직원은 현재 직장인 B사에서 퇴사할 계획을 가지고 있음.
- 그러던 중, B사의 최근 진행하고있는 사업의 수주요청서를 알게 됨.
- 퇴사 후 새로 회사를 설립하여 B사와 이 수주 건으로 경쟁하고 싶어함.
- 그러나, 수주 요청은 퇴사 이전에 마감될 것으로 전망됨.

해당 직원이 퇴사 이전에 이 수주 요청에 응답한다면, **고용주와 직접적인 경쟁관계**로 이어지므로 기준 4A를 위반하게 됨. 퇴사 등으로 이해관계를 청산하거나 B사 및 고객사의 사전 동의없이는 불가능함.

사례 5 : 내부 고발과 충성 의무의 상충 여부

- A는 투자 운용사의 트레이딩 데스크에서 근무하고 있으며, 헤지펀드를 운용 중.
- 최근 시황급변으로 인해 헤지펀드에 손실이 발생할 것으로 예상되었으나, 손실이 반영되지 않음.
- 이 사항에 대해 상급자와 컴플부서에 알렸으나, 관여하지 말라는 답변만 받게 됨.

명백한 업무상 오류를 발견하고, 내부고발 절차에 맞게 상급자와 컴플라이언스 부서에 알렸다면 기준 4A를 위반하지 않음. 이후 묵살된 의견은 내부고발 절차에 따라 진행되어야 할 것으로 보임,