

161819

오홍석 김민식 유광열

“주식 보험 서비스”

아이디어 제안서

INDEX

0. 예선 분석모형 설명

1. 아이디어 제안 배경

2. 주식 보험 서비스 수익구조

3. 주식 보험 서비스 Simulation with Data

4. 한계점 및 나아가야 할 방향

5. 결론

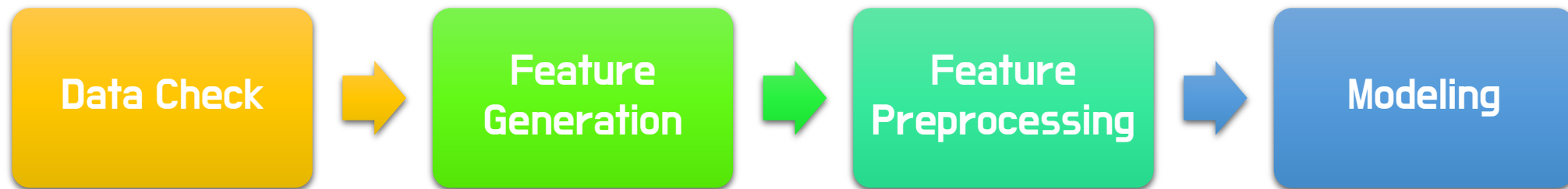
6. 참고자료

0

예선 분석 모형 설명

[분석 배경]

- 주식 보유기간을 예측하는데 있어서 여러 요인을 찾음: 기존 데이터와 MERGE의 어려움으로 사용하지 않음
- 다양한 관점으로 Feature 생성: 전진 선택법을 활용하여 Feature를 선택
- 여러 종류의 전처리 & Handling을 시도: label값을 잘 예측하는 방법들만 선택하여 적용
- 여러 방법으로 모델링을 진행: 이후 선택된 모델들 가지고 앙상블 진행

[분석 과정]

A. Data Check

- 다양한 EDA를 시도하여 데이터 구조 이해 및 특징 추출
- 논문 및 다양한 자료를 통해 외부 데이터 여부 확인
- 기존에 제공된 Base Line 이해 및 train hist_d 설정

B. Feature Generation

USE DATA

- ✓ CUS_INFO
- ✓ IEM_INFO
- ✓ STK_BNC_HIST
- ✓ STK_HLD_TRAIN

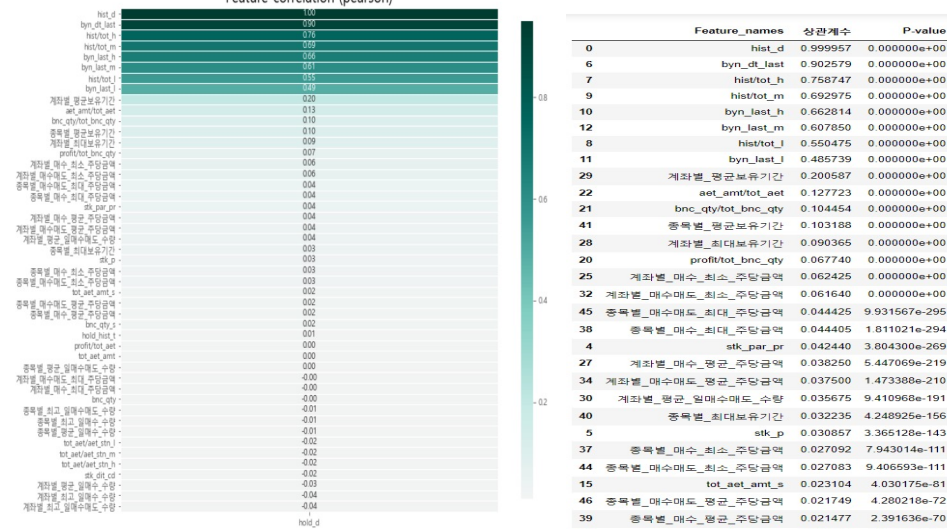


Base Feature	Base Generation Feature	STK HLD BASE FEATURE	
HIST_D	BYN_DT_LAST	매수 Feature	계좌 평균 일 매수수량
			계좌 매수평균 주당금액
SEX_DIT_CD	HIST/TOT_(H,L,M)	(계좌 별) 이전 주식 보유기간	계좌 평균보유기간
			계좌 최대보유기간
:	:	(계좌 별) 매수/매도 Feature	계좌 평균일매수매도 수량
			계좌 매수매도평균 주당금액
STK_PAR_PR	BNC_QTY/TOT_BNC_QTY	(주식 종목 별) 이전 주식 보유기간	종목별 최대보유기간
			종목별 평균보유기간
STK_P	AET_AMT/TOT_AET	(주식 종목별) 매수 매도 Feature	종목별 평균 일매수매도 수량
			종목별 매수매도평균 주당금액

BASE Feature + Feature Generation => 총 58개 Feature 사용

1. Correlation 확인 및 Statistical verification

Feature-correlation (pearson)

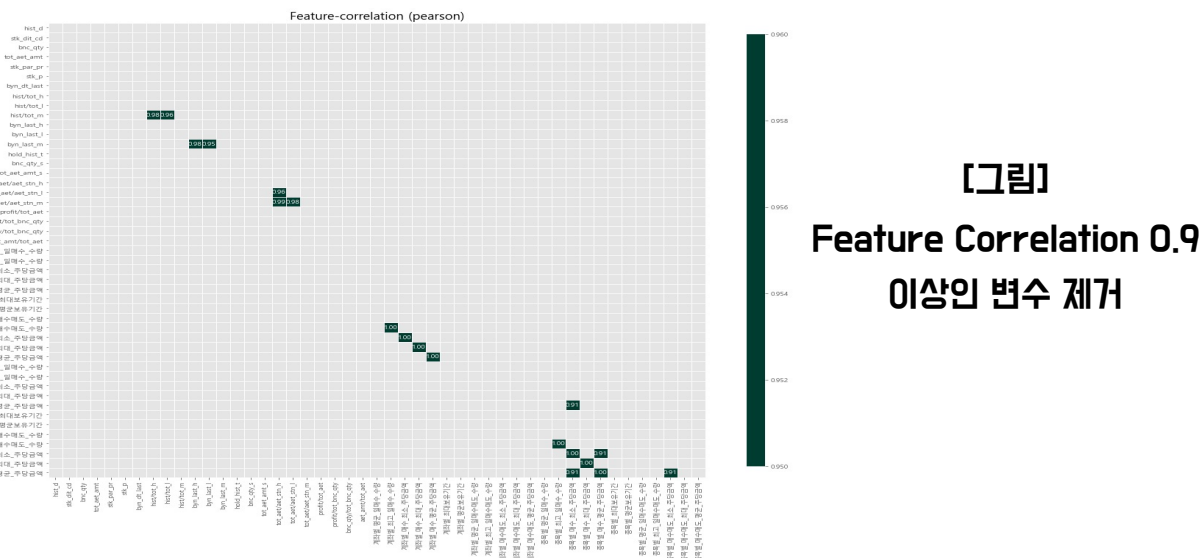


상관계수 0.9 넘는
변수 제거

[표] Continuous Correlation

[표] VIF 확인

2. Feature Correlation를 통해 Feature 제거

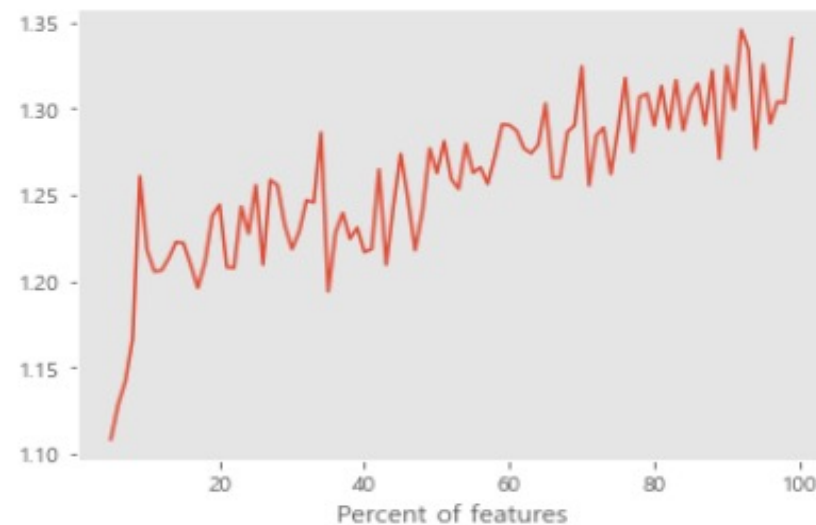


3. Category Feature One-Hot Encoding + PCA 적용

	pca_0	pca_1	pca_2	pca_3	pca_4	pca_5	pca_6	pca_7	pca_8	pca_9	pca_10	pca_11	pca_12	pca_13	pca_14
0	0.252054	0.493103	1.169701	0.323112	0.291077	-0.613599	-0.134122	0.241614	-0.304018	-0.002141	0.573250	0.733067	-0.502705	0.048057	-0.180382
1	0.252054	0.493103	1.169701	0.323112	0.291077	-0.613599	-0.134122	0.241614	-0.304018	-0.002141	0.573250	0.733067	-0.502705	0.048057	-0.180382
2	0.271667	0.567320	1.005716	0.270820	0.188129	0.356687	0.517413	-0.479833	-0.218878	0.148161	0.735733	0.698621	-0.523450	0.080364	-0.187228
3	0.271667	0.567320	1.005716	0.270820	0.188129	0.356687	0.517413	-0.479833	-0.218878	0.148161	0.735733	0.698621	-0.523450	0.080364	-0.187228
4	0.307229	0.853838	0.194931	0.140299	0.705785	-0.558155	0.164987	-0.481071	-0.268952	0.196665	0.735933	0.632021	-0.540569	0.063290	-0.187239

[표] Category Feature One-hot Encoding한 값을 PCA 적용한 결과

4. Feature Selection을 변수 선택



[그림]
Extra Regression을 통해
Feature Selection 진행

1. Modeling

모델	K-FOLD validation 평균 RMSE
KNeighborsRegressor	56.198
ExtraTreesRegressor	0.475
GradientBoostingRegressor	0.564
XGBRegressor	0.441
LGBMRegressor	0.892
CatBoostRegressor	0.962

2. Averaging Ensemble

2개 모델 조합	K-FOLD validation 평균 RMSE
ExtraTreesRegressor●GradientBoostingRegressor	0.476
ExtraTreesRegressor●XGBRegressor	0.441
3개 모델 조합	K-FOLD validation 평균 RMSE
ExtraTreesRegressor●GradientBoostingRegressor ●XGBRegressor	0.451
ExtraTreesRegressor●GradientBoostingRegressor ●LGBMRegressor	0.542
4개 모델 조합	K-FOLD validation 평균 RMSE
ExtraTreesRegressor●GradientBoostingRegressor ●XGBRegressor●LGBMRegressor	0.497
ExtraTreesRegressor●GradientBoostingRegressor ●XGBRegressor●LGBMRegressor	0.507

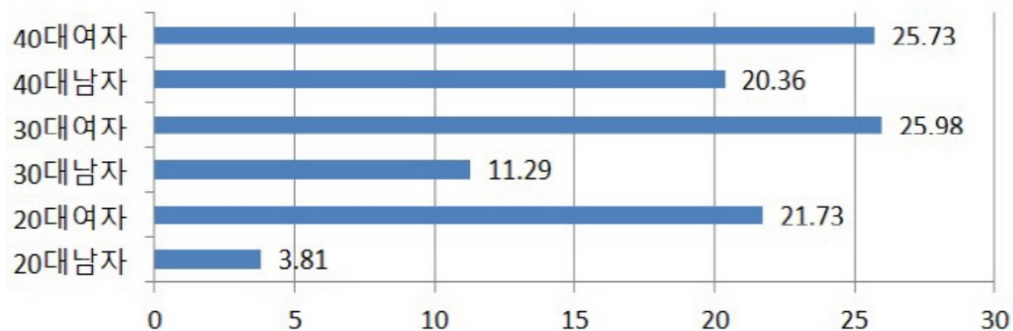
최종 Model : ExtraTrees Regressor * XGB Regressor
최종 Score : 0.553(validation) / Public : 63.5662, Private : 63.269

1

아이디어 제안 배경

‘주식인구 1천만’ 시대...20·30대 소액투자대열 이어져

작년 세대, 성별 신규
주식계좌 수익률(%)



개인투자자, 10명 중 4명 주식으로 손실... 1명만 연 1000만원 초과 수익

개인투자자 주식 양도차익 구간별 인원 추정치

	손실	0원~ 1000만원	1000만~ 2000만원	2000만원 초과	총
비율	약 40%	약 50%	약 5%	약 5%	약 100%
인원 (명)	약 240만명	약 300만	약 30만	약 30만	약 600만

주식 계좌 수 ↑

20.30 대 투자 인구 ↑

개인투자자 주식 손실 비율 낮지 않음



주식 거래
보험 서비스

주식 거래 보험서비스 모델

[NH 투자증권]

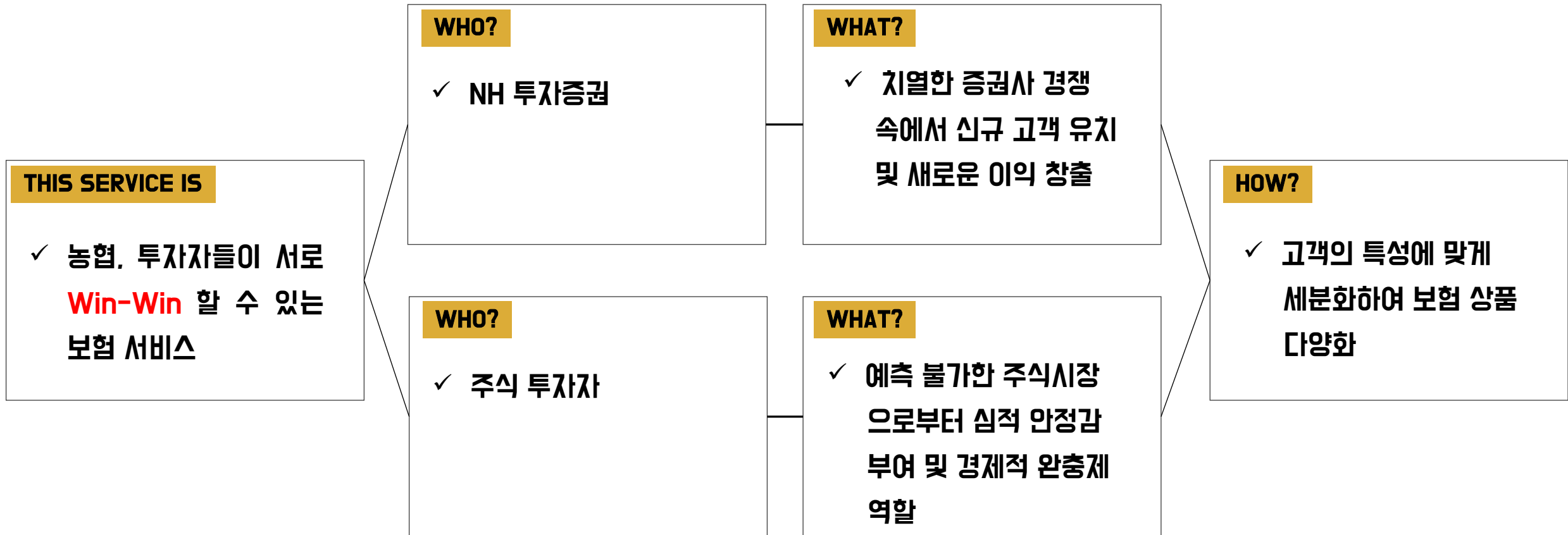
- 보험 상품 통한 신규 투자자 유입 및 기존 투자자 유지
- 수수료 이외 새로운 방식인 "보험료"를 통한 수익 창출

[실행 가능성]

- 주식 투자자들의 근본적인 두려움인 "손해"에 대한 접근
- 수 많은 투자자의 주식거래 정보 데이터의 활용 가능

[고객]

- 손해 발생 시 일정 금액의 보험금 수령 가능
- 주식거래에 있어서 보험이 주는 심리적 안정감



2

주식 보험

서비스 수익구조

[회사의 수익]

① 고객에게 납부 받은 보험료의 환불 비율의 나머지를 수익으로 산정

- 주식 판매 시 수익이 나면 고객 보험 환급 비율 증가 → 회사가 받는 보험료 수익 감소
- 주식 판매 시 손해가 나면 고객 보험 환급 비율 감소 → 회사가 받는 보험료 수익 증가

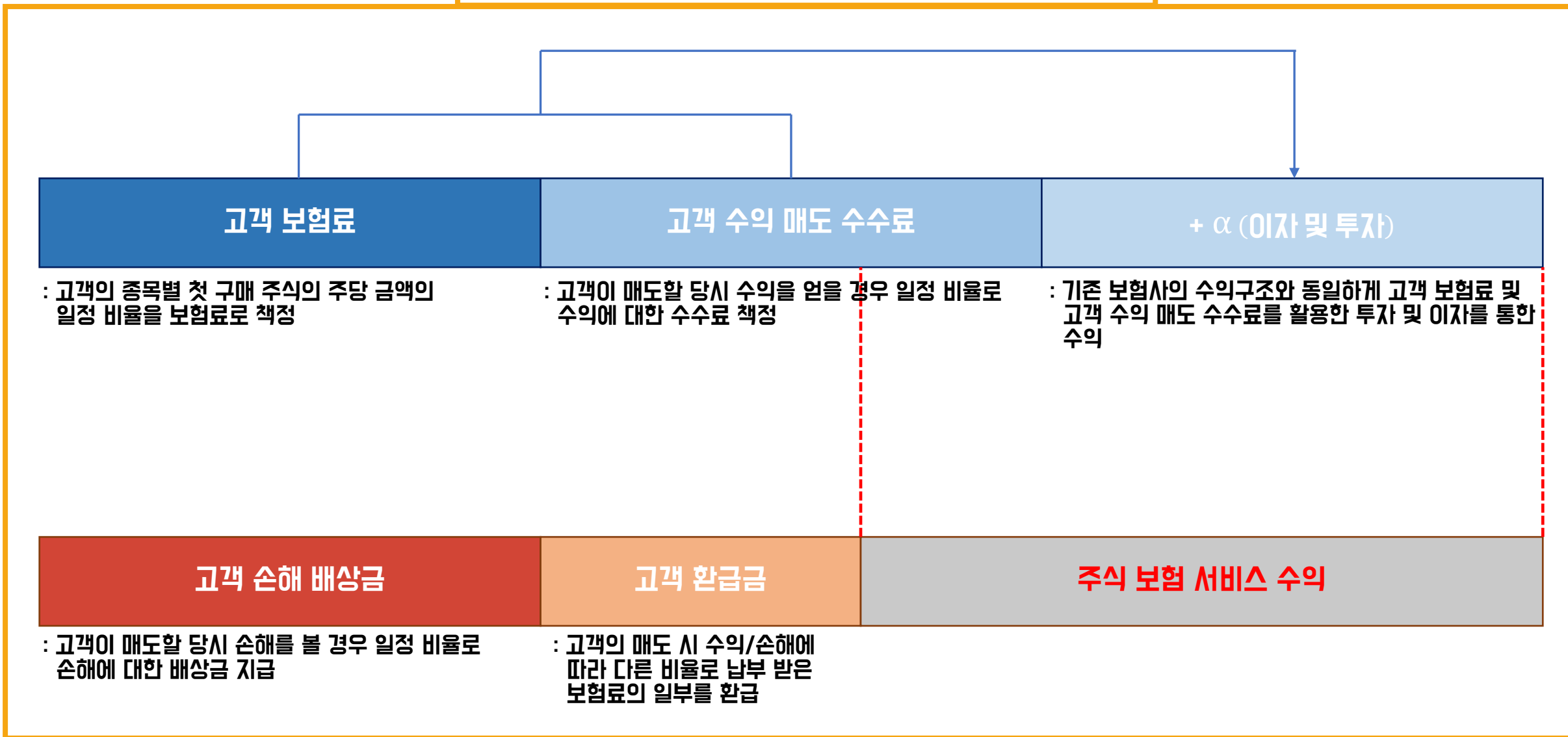
② 고객이 주식 판매 시 수익이 나면 일정부분 수수료를 수익으로 산정

③ : ①, ② 과정에서 생기는 수익을 활용한 투자 및 이자

[회사의 비용]

① 고객이 주식 판매 손해 발생 시 배상금

주식 보험 서비스 기본 수익모델



- 주식 보험 서비스 모델 Simulation에서 판매금액 필요하기에 stk_bnc_hist 데이터를 활용

→ stk_bnc_hist 데이터에서 [1] 계좌/종목별 주식 구매를 분리 후 [2] 두 번 이상의 판매 기록이 남아있는 데이터를 전처리 후 수집

	계좌번호	기준일자	종목코드	주당금액	종목보유잔고수량	종목보유잔고수량_shift	전체종목평가금액	전체종목평가금액_shift	일_매수_매도_수량	계좌별_종목별_총보유기간_real	해당일시_주당금액
0	0000c7a5a5a1db9c7e890da984b20d544c09cb6f51c546...	2019-03-21	A002270	5000.0	1.0	0.0	667000.0	0.0	1.0	351	667000.0
1	0000c7a5a5a1db9c7e890da984b20d544c09cb6f51c546...	2019-03-28	A002270	5000.0	2.0	1.0	1270000.0	667000.0	1.0	351	635000.0
2	0000c7a5a5a1db9c7e890da984b20d544c09cb6f51c546...	2019-05-07	A002270	5000.0	3.0	2.0	1815000.0	1270000.0	1.0	351	605000.0
3	0000c7a5a5a1db9c7e890da984b20d544c09cb6f51c546...	2019-07-16	A002270	5000.0	4.0	3.0	2104000.0	1815000.0	1.0	351	526000.0
4	0000c7a5a5a1db9c7e890da984b20d544c09cb6f51c546...	2020-01-29	A002270	5000.0	8.0	4.0	2964000.0	2104000.0	4.0	351	370500.0
5	0000c7a5a5a1db9c7e890da984b20d544c09cb6f51c546...	2020-02-07	A002270	5000.0	10.0	8.0	3565000.0	2964000.0	2.0	351	356500.0
6	0000c7a5a5a1db9c7e890da984b20d544c09cb6f51c546...	2020-07-08	A002270	5000.0	8.0	10.0	2828000.0	3565000.0	-2.0	351	353500.0
7	0000c7a5a5a1db9c7e890da984b20d544c09cb6f51c546...	2020-08-18	A002270	5000.0	0.0	8.0	0.0	2828000.0	-8.0	351	353500.0

— 구매
— 판매1
— 판매2

[그림] Hist Data Frame 모습

- 가정 1. "전체종목 평가금액" 0일 시 판매금액을 알 수 없기에, 가장 최근(직전) 시점의 주당 금액을 사용해 계산
- 가정 2. 주식을 최초 매수 이후 "매수-매도"가 여러 번 등장하는 경우 계산상 편의를 위해, "종목보유 잔고수량"이 최대 이후에 전부 매도하였다고 가정

주식 보험 서비스 기본 수익모델 변수

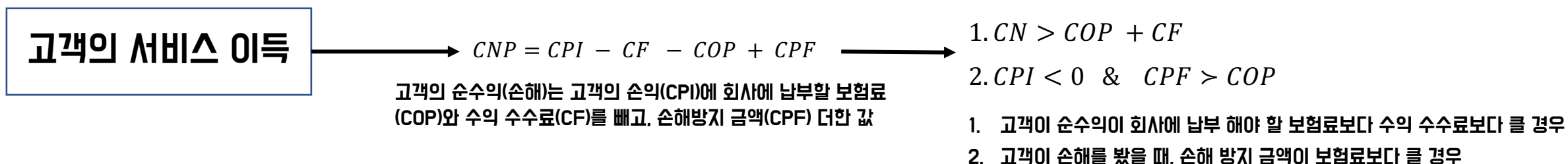
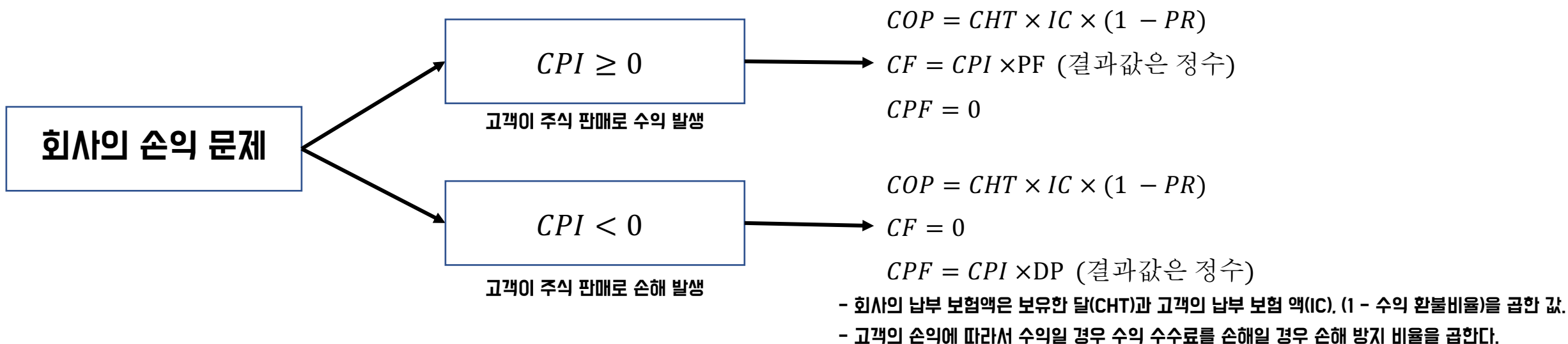
변수명	변수 설명
ICR (고객 보험료 비율)	새로운 종목의 주식 거래 시 주식의 한 주 가격에 적용할 납입보험료 수수료 비율
PR (수익 환급 비율)	고객이 수익을 낼 시 돌려줄 환급액 비율
LR (손해 환급 비율)	고객이 손해를 볼 시 돌려줄 환급액 비율
PF (수익 수수료 비율)	고객이 수익을 낼 시 납입해야 할 수수료 비율
DF (손해 배상금 비율)	고객이 손해를 볼 시 방지해줄 배상금 비율

Simulation 함수의
Input 변수
(비율)

변수명	변수 설명
cpl (고객손익)	(최대보유 주당금액 - 판매금액) x 최대보유수량
cht (보유 개월 수)	$\text{Int}(\text{보유기간} / 30) + 1$: 한달이 넘지 않아도 보험료 납입
IC (고객 납부 보험료)	최초 매수 시 주당 금액 x ICR x 최대보유수량
cop (회사가 얻는 보험료)	고객이 수익을 낼 시 납입해야 할 수수료 비율
cpf (손해 방지 배상금)	고객이 손해를 볼 시 방지해줄 배상금 비율
cnp (고객 순 수익/손해)	고객 손익과 손해 방지 금액에서 회사에 납부해야 할 비용(보험료/수수료) 차이

Simulation 함수
내부 계산을
위한 변수

CPI	IC	COP	CPF	CNP	CHT
고객 손익	고객이 납부할 보험료	회사가 얻는 보험료	손해 방지 배상금	고객의 순 수익/손해	보유한 기간 (달)



3

주식 보험 Simulation with Data

고객의 특징 별로 세분화된 다양한 보험상품을 만들어 Simulation을 진행하기 위해 **군집화**를 진행



위에서 생성한 데이터를 바탕으로 주식 보험 신청 여부를 0/1 값을 **Random** 하게 부여



군집의 종류에 따라 다른 값의 변수로 함수 실행 후

1. NH 투자증권 사의 이익

2. 얼마나 많은 고객이 이득을 봤는지

계산

[K-means clustering]

주어진 데이터를 k개의 클러스터로 묶는 알고리즘
각 클러스터 거리 차이의 분산을 최소화하는 방식으로 동작

$$V = \sum_{i=1}^k \sum_{x_j \in S_i} |x_j - \mu_i|^2$$

μ_i : i 번째 클러스터의 중심 S_i : 클러스터에 속하는 점의 집합

전체의 분산을 계산

- (1) 주어진 데이터로부터 k개의 중심점을 결정
- (2) 중심점과 모든 데이터를 유클리드 거리로 계산
- (3) 해당 데이터에 가장 가까운 새로운 중심점을 찾음

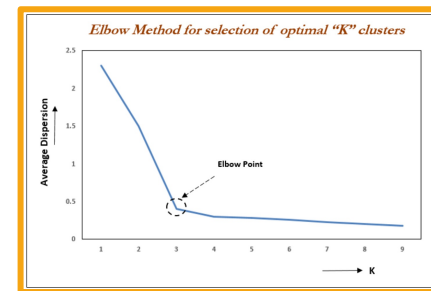
[silhouette coefficient]

군집 간의 거리가 효율적으로 분리되어 있는지 보여주는 지표
동일 군집은 가깝게, 타 군집과는 멀리 떨어져 있음을 의미

$$s(i) = \frac{(b(i) - a(i))}{(\max(a(i), b(i)))}$$

[Elbow Method]

클러스터 간 거리의 합이 급격히 떨어지는 구간을 군집의 개수로 선택하는 방법



[그림] Elbow 예시

(전처리 된) stk_bnc_hist 데이터를 사용하여 계좌/종목별 feature 생성

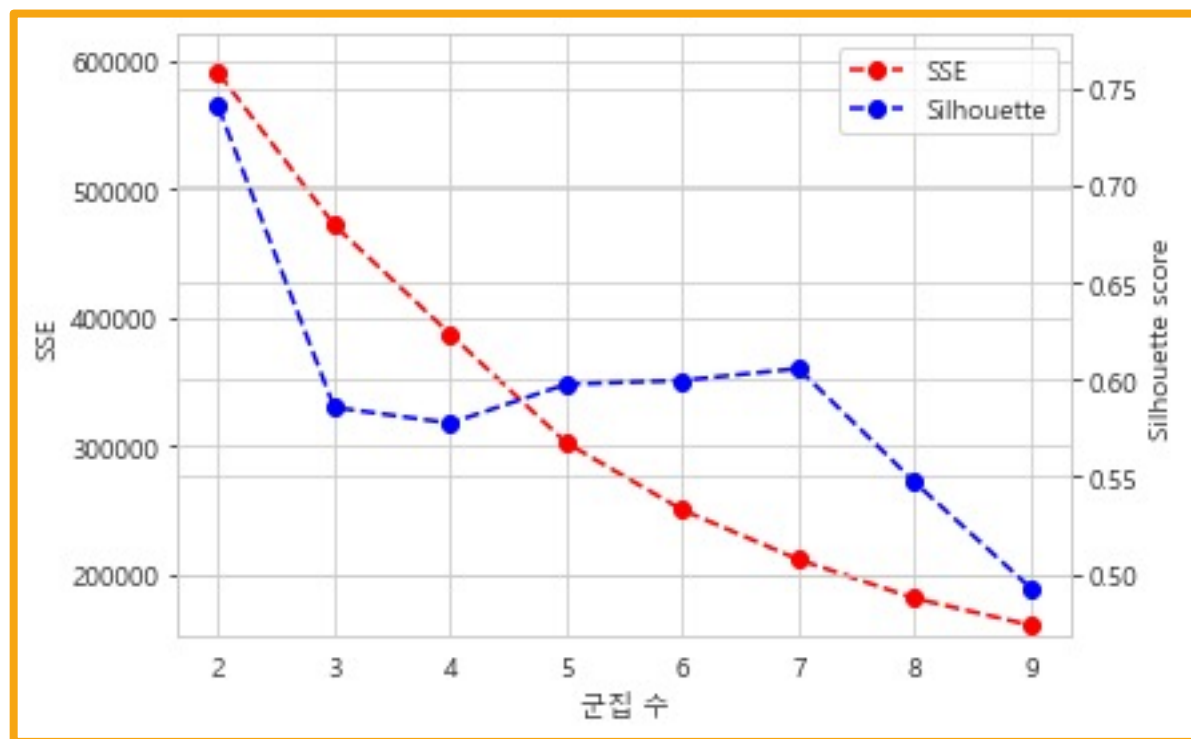
		계좌	종목	보유기간	최대보유수량	최대보유개당금액	판매금액	최초구매주당금액
0	0000c7a5a5a1db9c7e890da984b20d544c09cb6f51c546...	A002270		351.0	10.0	356500	353500.0	667000.0
1	0000c7a5a5a1db9c7e890da984b20d544c09cb6f51c546...	A037070		3.0	30.0	5620	7300.0	5620.0
2	0000c7a5a5a1db9c7e890da984b20d544c09cb6f51c546...	A051360		47.0	40.0	6910	7540.0	6910.0
3	0000c7a5a5a1db9c7e890da984b20d544c09cb6f51c546...	A060980		25.0	11.0	43000	43800.0	40800.0
4	0000c7a5a5a1db9c7e890da984b20d544c09cb6f51c546...	A065450		25.0	31.0	3165	3165.0	3310.0
...
174075	fff7d563e388744fdc1ca90153ec0161d1345511cb9070...	A038070		15.0	1051.0	12700	14250.0	12700.0
174076	fff7d563e388744fdc1ca90153ec0161d1345511cb9070...	A065500		8.0	14854.0	930	874.0	800.0
174077	fff7d563e388744fdc1ca90153ec0161d1345511cb9070...	A083500		16.0	672.0	10150	9380.0	10150.0
174078	fff7d563e388744fdc1ca90153ec0161d1345511cb9070...	A093230		33.0	17324.0	277	360.0	277.0
174079	fff9cf3aaffd32dcbf70dce8e5216411f9d38537b477d6...	A005930		98.0	35.0	55400	59200.0	49850.0

174080 rows × 8 columns

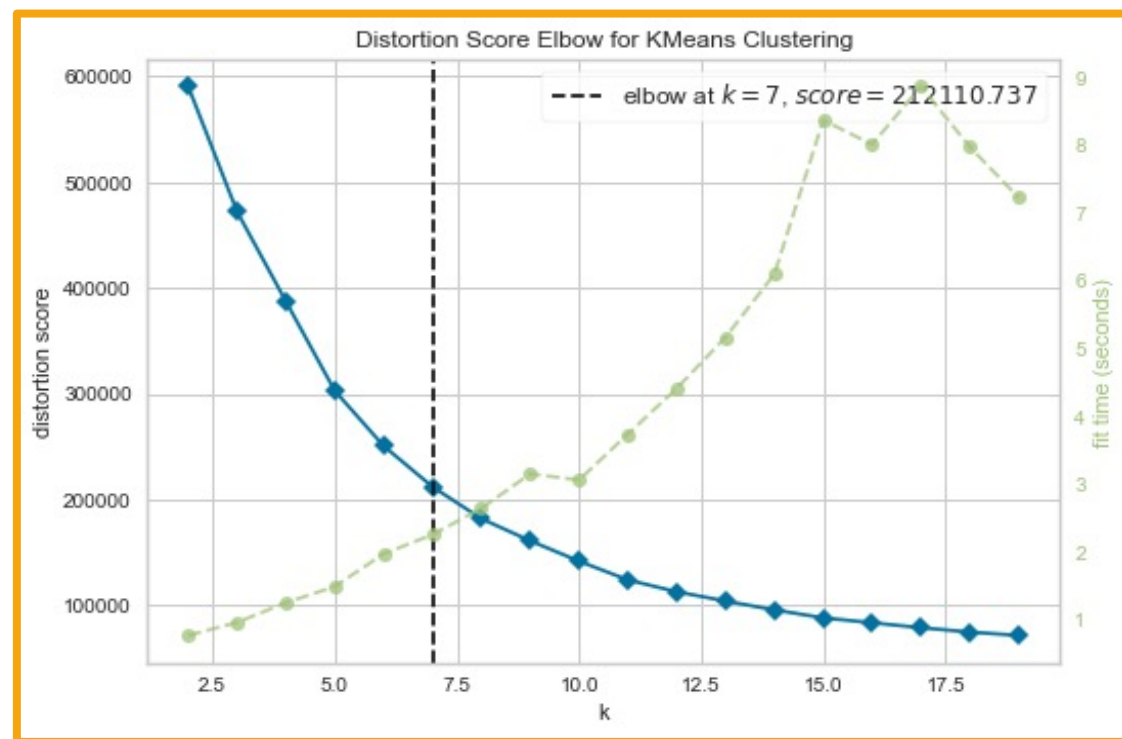
[그림] (군집화를 위해) 계좌-주식 별 만든 Feature Data Frame

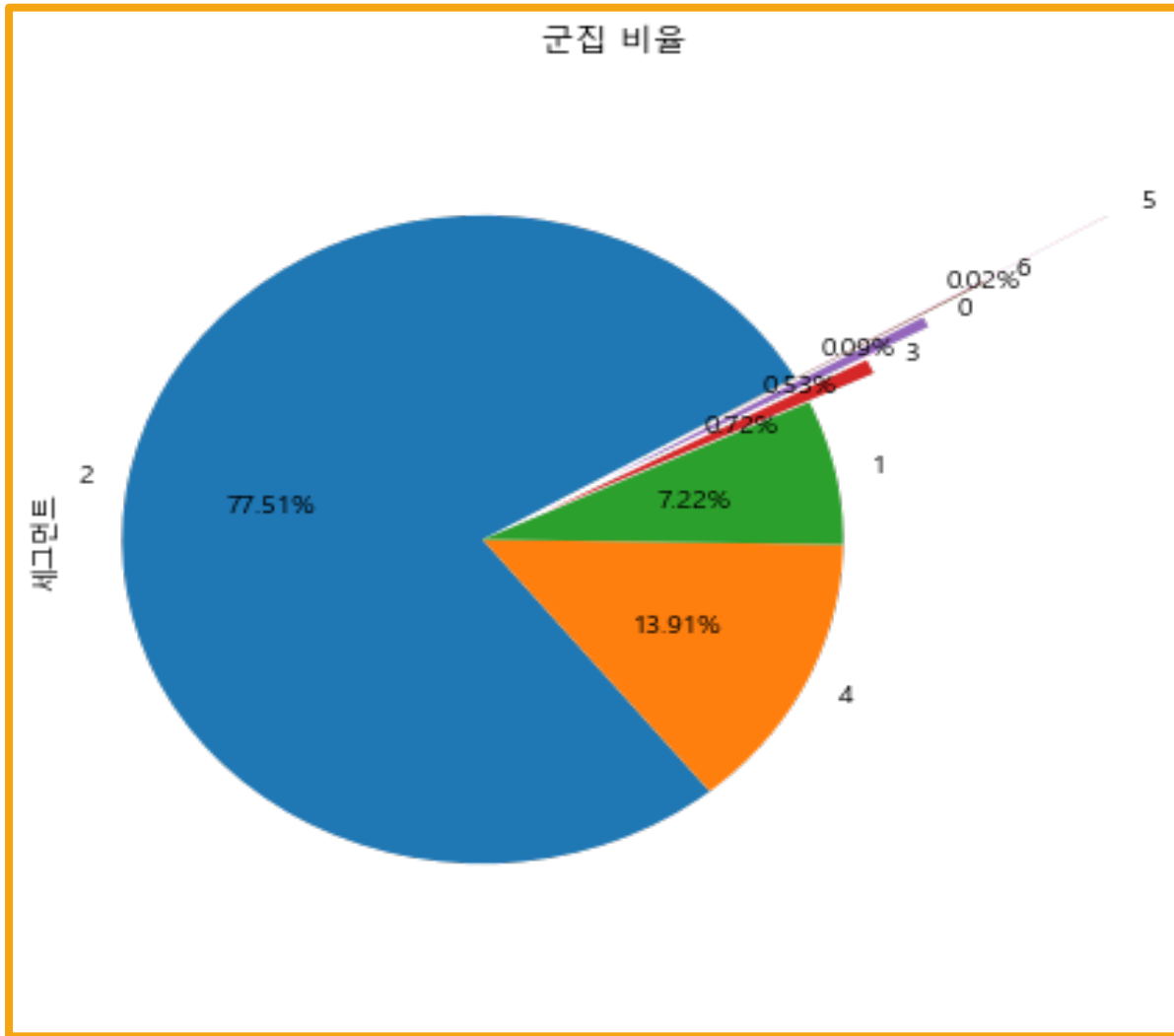
Feature 명	의미
보유기간	고객이 해당 주식 종목을 보유한 기간
최대보유수량	고객이 각 주식 종목별로 보유한 최대수량
최대보유 개당 금액	고객이 각 주식 종목별로 최대 보유 수량일 때 한 주의 금액
판매금액	(앞선 가정한 가설을 바탕으로 만들어진) 각 주식 종목별 판매금액
최초구매 주당 금액	고객이 각 주식 종목별 최초로 구매할 때 한 주의 금액
군집화 진행 전 각 Feature마다 표준화 (Standard Scaling 진행)	

- SSE 값이 가장 평할 해지고, 실루엣 score가 가장 높은 7개로 군집 수를 결정
- Python 내부의 `kelbow_visualizer`에서도 동일하게 $k=7$. 군집이 가장 최적의 군집으로 판단



[표] 군집 개수를 선택하기 위한 실루엣 계수 및 SSE

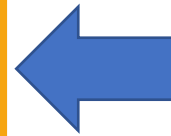
[표] `kelbow_visualizer`



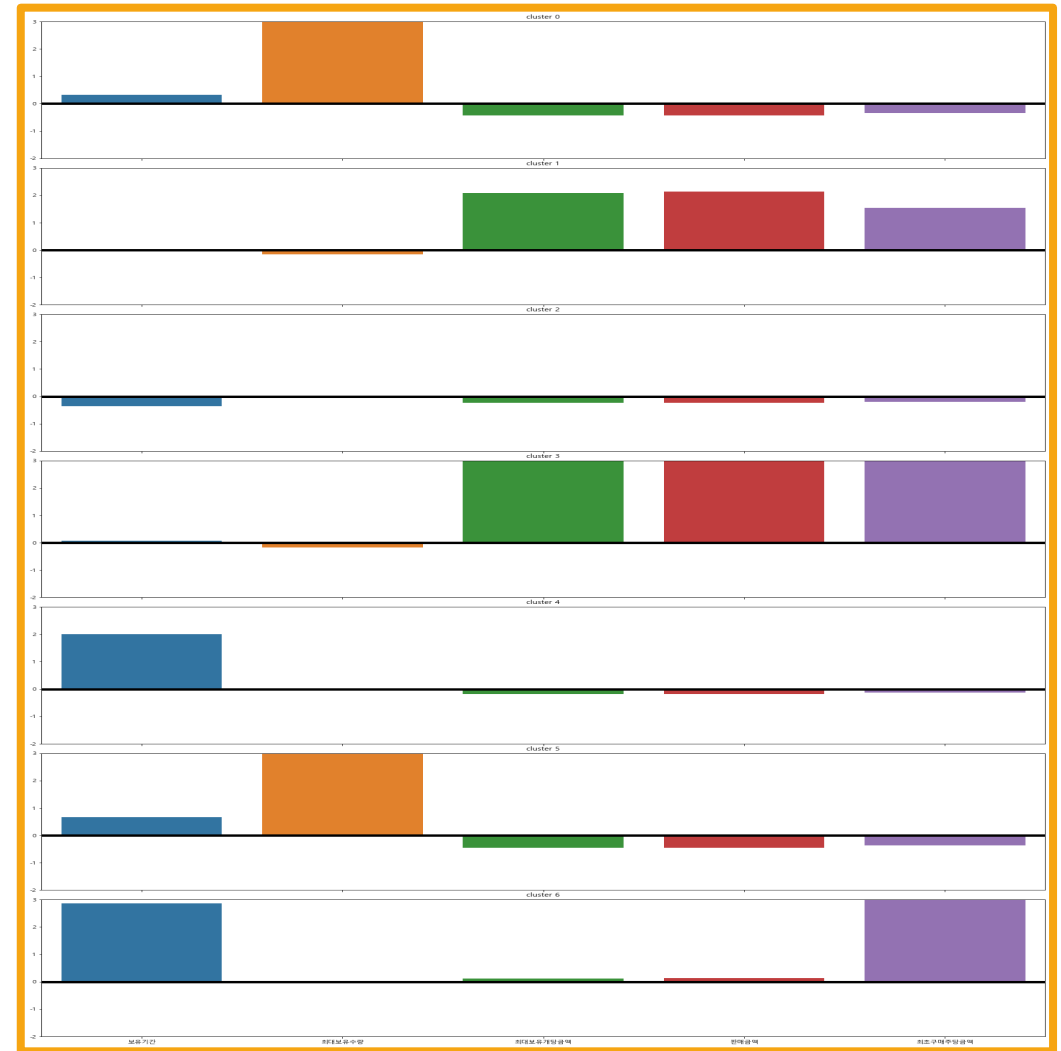
[표] 군집 이후, 군집 별 분포(비율)

0번 군집 : 0.53% / 1번 군집 : 7.22% / 2번 군집 : 77.51% / 3번 군집 : 0.72%

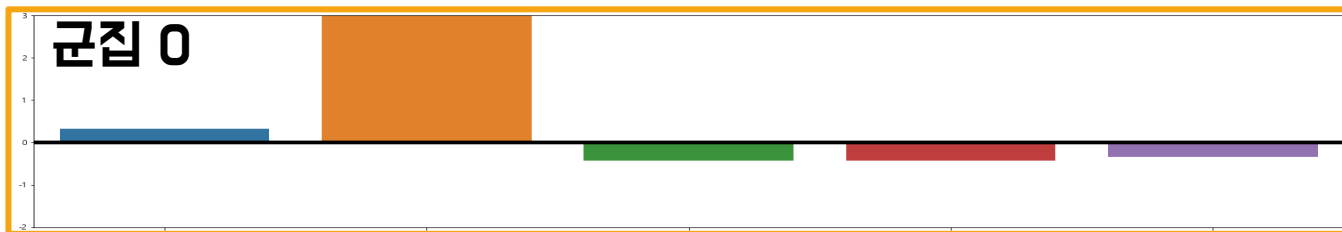
4번 군집 : 13.91% / 5번 군집 : 0.02% / 6번 군집 : 0.09%



각 군집 별
비율
(분포)

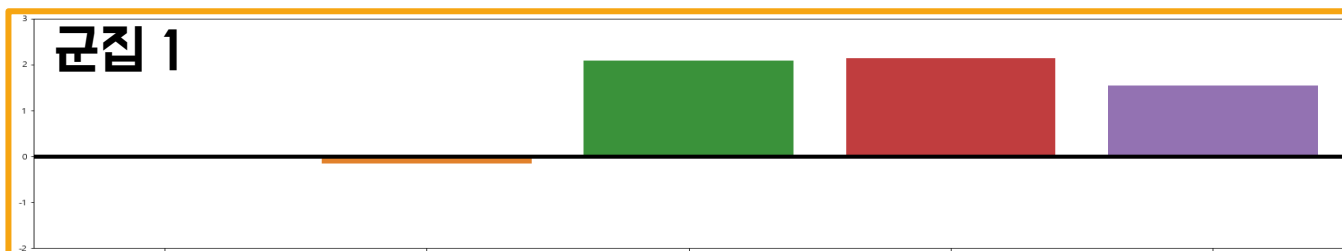


[표] 군집 별 특성



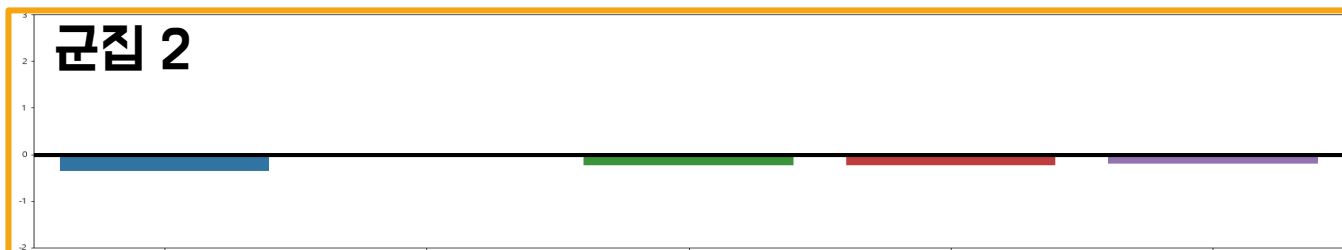
[군집 0]

- 특징: 최대 보유수량이 높음
- 해석: 초기 투자 비용이 낮고, 평균 주 가격이 낮은 주식(소형주) 대량 매수하는 군집(상대적으로 긴 보유기간)



[군집 1]

- 특징: 최소 구매 주당 금액 + 최대 보유 개당 금액이 높음
- 해석: 평균적인 보유기간을 가지며, 초기 주식 투자비용이 높고, 주당 금액이 높은 주식(대형주)를 보유한 군집



[군집 2]

- 특징: 모든 Feature 평균 이하
- 해석: 보유기간도 상대적으로 짧고, 초기 투자 비용, 가격이 낮은 집단 (소량의 주식)을 단기간 매도 및 매수하는 군집으로 예측



[군집 3]

- 군집 1과 매우 유사한 군집
- 상대적으로 군집 1의 주요 특성(금액) 대비 높음

보유기간 최대보유 수량 최대보유 개당 금액 판매금액 최초 구매 주당 금액



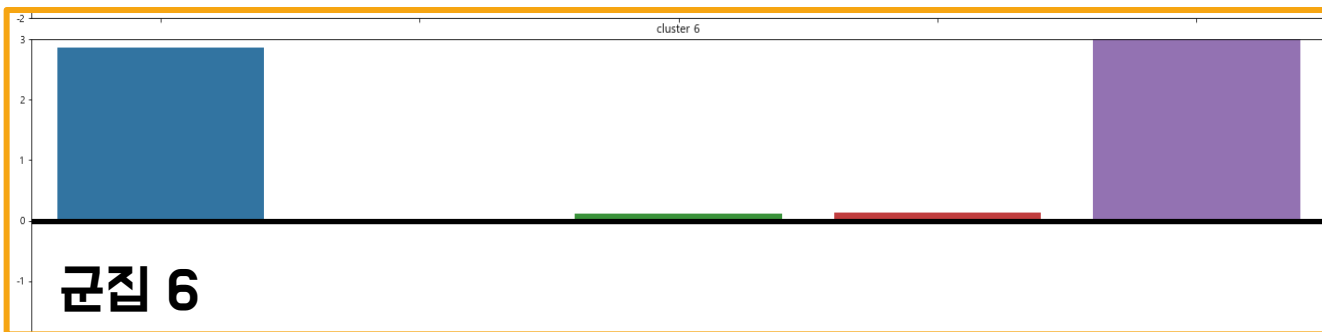
[군집 4]

- 특징: 보유기간이 약간 높음
- 해석: 주당 금액이 적은 주식을 장기간 보유한 사람들 군집



[군집 5]

- 군집 0과 매우 유사한 군집



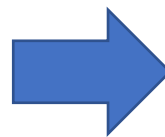
[군집 6]

- 특징: 보유기간, 최초 구매 주당 금액이 상대적으로 매우 높음
- 해석: (상대적으로 최초 구매 주당 금액이 높기 때문에) 대형주 혹은 안정성에 대하여 구매하여 **장기간 보유한 사람들의 주식 군집**

보유기간 최대보유 수량 최대보유 개당 금액 판매금액 최초 구매 주당 금액

[1] 고객이 주식에 대한 보험 신청

	계좌	종목	보유기간	최대보유수량	보험신청
0	01200ecc1f065e2826b4babb4acdfc22a163f67910a9b8...	A031820	41.0	54526.0	0
1	01200ecc1f065e2826b4babb4acdfc22a163f67910a9b8...	A036120	107.0	44034.0	0
2	027abcc3ff4d40bab915b746d80554c37c765a20cbadbb...	A065940	7.0	29396.0	0
3	02b996de6614fc0ba8c87df266ca07bffe945bf8fd598...	A001340	648.0	115000.0	0
4	02b996de6614fc0ba8c87df266ca07bffe945bf8fd598...	A002150	92.0	43207.0	0
...
911	ffdccfb9dc24cf41843f6e7f34704ccb73ac81acc29aa3...	A104480	8.0	28002.0	1
912	ffdccfb9dc24cf41843f6e7f34704ccb73ac81acc29aa3...	A122800	6.0	61360.0	1
913	ffdccfb9dc24cf41843f6e7f34704ccb73ac81acc29aa3...	A204990	249.0	38953.0	0
914	ffdccfb9dc24cf41843f6e7f34704ccb73ac81acc29aa3...	A222810	339.0	34076.0	0
915	ffdccfb9dc24cf41843f6e7f34704ccb73ac81acc29aa3...	A223310	90.0	55004.0	1

Simulation
Function

Input Variation 조합

[2] 보험 신청한 경우 회사/고객 수익 파악

	회사이산손익	회사손익금액	고객이산손익	고객순손익	회사손익누적금액
0	1	96522	0	-4587990	96522
1	1	1002747	1	100101632	1099269
2	1	462214	0	-12504832	1561483
3	1	767280	1	19632720	2328763
4	1	33240	0	-33240	2362003
...
379	1	7719	0	-7719	85512045
380	1	131658	0	-131658	85643703
381	0	-763	0	-314371	85642940
382	1	5816	0	-5816	85648756
383	1	218760	0	-218760	85867516

- 보험 신청자를 계좌/종목에 따라서 Random하게 부여

1) " $p = 0.4$ " 따르는 (이항분포) 무작위 표본 추출

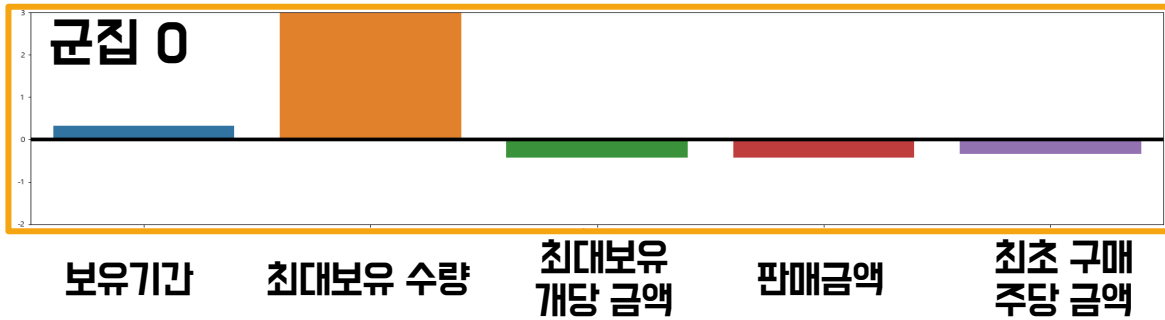
2) 보험 신청여부: 1 - 신청 / 0 - 신청하지 않음

- 보험 신청한 고객/주식에 대해 Simulation 함수 적용

1) 개별 보험 신청 건에 대한 회사 고객의 손해/수익 파악

2) 고객과 회사 전체의 수익/손실 금액, 수익 비율 파악

1. 군집 선택



- 최대 보유 수량 대비 개당 금액이 낮으므로 다른 군집에 비해 **소형주를 많이 매입한 군집**으로 판단
- 박리다매 형식으로 거래하는 것을 목표(수수료, 보험료 상대적으로 저렴)

2. Function input 요소 선택

Feature 명	Input 값(비율)
ICR(납입 보험료 비율)	0.5%, 1%, 2%, 2.5%, 5%
PR(수익 환급 비율)	90%, 95% 97.5% 99%
LR(손해 환급 비율)	
PF(수익 수수료 비율)	0.5%, 1.0%, 2.0%
DF(손해 배상금 비율)	0.5%, 1.0%, 2.5%, 5.0% 10.0%, 25.0%, 50.0%

[표] Simulation Function 요소들 값 예시

- 환급 비율 / 배상금 비율 높고 다양하게 진행
- 납입 보험료/수수료 상대적으로 낮게 진행

3. Simulation Function input 요소별 결과 확인

```
print('전체 거래 수: ', Pal_bas_df_0.shape[0])
print('보험 신청 거래 수: ', Pal_bas_df_0.shape[0] - len(Pal_df))
print('이득본 고객 수: ', sum(Pal_df.고객이산손익))
print('이득본 고객 수 비율: ', sum(Pal_df.고객이산손익)/len(Pal_df.고객이산손익))
```

전체 거래 수: 916
 보험 신청 거래 수: 556
 이득본 고객 수: 150
 이득본 고객 수 비율: 0.4166666666666667

```
print('보험 신청 거래 수: ', Pal_bas_df_0.shape[0] - len(Pal_df))
print('이득본 회사 거래 수: ', sum(Pal_df.회사이산손익))
print('회사손익누적금액: ', Pal_df.iloc[-1, -1])
print('이득본 회사 거래 수 이득비율: ', sum(Pal_df.회사이산손익)/len(Pal_df.회사이산손익))
```

보험 신청 거래 수: 556
 이득본 회사 거래 수: 299
 회사손익누적금액: 190323984
 이득본 회사 거래 수 이득비율: 0.8305555555555556

```
print('회사 총 이득 금액: ', sum(Pal_df.query('회사이산손익 == 1')['회사손익금액']))
print('회사 총 손해 금액: ', sum(Pal_df.query('회사이산손익 == 0')['회사손익금액']))
print('고객 총 이득 금액: ', sum(Pal_df.query('고객이산손익 == 1')['고객손익금액']))
print('고객 총 손해 금액: ', sum(Pal_df.query('고객이산손익 == 0')['고객손익금액']))
```

회사 총 이득 금액: 258793836
 회사 총 손해 금액: -68469852
 고객 총 이득 금액: 3879618232
 고객 총 손해 금액: -9539759090

4. (3번 과정을 자동화) 회사/고객의 보험 수익 정보 Excel 정리

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1																								
2		납입 보험료	수익 환급 비율	손해 환급 비	수익 수수료	손해 배상지			전체 거래수	보험 신청수	보험 신청비율	이득 본 고객수	이득 본 고객비율	이득 본 회사 거래수	이득 본 회사 거래수 비율		회사 총 수익액	회사 총 손실액	수익/손실 금액(회사)		고객 총 수익액	고객 총 손실액	수익/손실 금액(고객)	
3		0.5%	90.0%	90.0%	0.5%	0.5%			916	370	40.4%	146	39.5%	317	85.7%		257,584,848	-22,985,530	234,599,318		3,932,466,704	-12,251,979,116	-8,319,512,412	
4		0.5%	90.0%	90.0%	0.5%	1.0%			916	376	41.0%	146	38.8%	288	76.6%		239,168,822	-64,252,560	174,916,262		3,578,174,768	-12,352,869,098	-8,774,694,270	
5		0.5%	90.0%	90.0%	0.5%	2.5%			916	359	39.2%	149	41.5%	251	69.9%		180,696,330	-236,771,052	-56,074,722		3,967,360,244	-11,972,730,628	-8,005,370,384	
6		0.5%	90.0%	90.0%	0.5%	5.0%			916	367	40.1%	157	42.8%	241	65.7%		153,585,489	-472,748,610	-319,163,121		4,358,026,299	-11,285,617,588	-6,927,591,289	
7		0.5%	90.0%	90.0%	0.5%	10.0%			916	342	37.3%	131	38.3%	203	59.4%		141,458,342	-776,667,571	-635,209,229		4,008,678,572	-9,455,333,573	-5,446,655,001	
8		0.5%	90.0%	90.0%	0.5%	20.0%			916	380	41.5%	153	40.3%	216	56.8%		121,250,108	-2,797,685,999	-2,676,395,891		3,510,756,191	-17,484,541,840	-13,973,785,649	
9		0.5%	90.0%	90.0%	0.5%	50.0%			916	357	39.0%	143	40.1%	190	53.2%		159,760,615	-5,412,276,660	-5,252,516,045		4,403,627,462	-16,671,492,570	-12,267,865,108	
10		0.5%	90.0%	90.0%	1.0%	0.5%			916	327	35.7%	132	40.4%	279	85.3%		194,974,496	-17,978,219	176,996,277		3,001,536,718	-8,645,860,797	-5,644,324,079	
11		0.5%	90.0%	90.0%	1.0%	1.0%			916	354	38.6%	154	43.5%	285	80.5%		265,879,272	-44,065,048	221,814,224		3,201,536,456	-9,087,664,154	-5,886,127,698	
12		0.5%	90.0%	90.0%	1.0%	2.5%			916	352	38.4%	145	41.2%	252	71.6%		245,037,983	-180,404,046	64,633,937		3,997,315,266	-9,984,631,721	-5,987,316,455	
13		0.5%	90.0%	90.0%	1.0%	5.0%			916	366	40.0%	154	42.1%	229	62.6%		233,256,021	-549,606,178	-316,350,157		4,141,030,887	-13,021,745,926	-8,880,715,039	
14		0.5%	90.0%	90.0%	1.0%	10.0%			916	380	41.5%	165	43.4%	232	61.1%		172,906,453	-1,048,641,443	-875,734,990		3,414,610,916	-12,543,410,250	-9,128,799,334	

5. Simulation 결과 : 회사의 총 수익 극대화(상위 10)

납입 보험액	수익 환불 비	손해 환불 비	수익 수수	손해 손해받	전체 거래수	보험 신청수	보험 신청비율	이득 본 고객수	이득 본 고객비율	이득 본 회사 거래수	이득 본 회사 거래수 비율	회사 총 수익액	회사 총 손실액	수익/손실 금액(회사)	고객 총 수익액	고객 총 손실액	수익/손실 금액(고객)
5.0%	90.0%	90.0%	0.5%	0.5%	916	352	38.4%	105	29.8%	346	98.3%	2,876,224,161	-4,697,333	2,871,526,828	3,759,820,193	-13,223,618,195	-9,463,798,002
5.0%	90.0%	90.0%	0.5%	5.0%	916	368	40.2%	111	30.2%	328	89.1%	2,399,154,874	-184,911,606	2,214,243,268	2,107,840,664	-12,244,254,171	-10,136,413,507
5.0%	90.0%	90.0%	1.0%	0.5%	916	385	42.0%	123	31.9%	380	98.7%	2,557,628,725	-2,243,161	2,555,385,564	2,622,195,844	-14,491,679,619	-11,869,483,775
5.0%	90.0%	90.0%	1.0%	1.0%	916	328	35.8%	112	34.1%	319	97.3%	2,058,383,293	-18,636,748	2,039,746,545	4,026,240,046	-10,051,015,849	-6,024,775,803
5.0%	90.0%	90.0%	1.0%	2.5%	916	369	40.3%	114	30.9%	340	92.1%	2,227,489,204	-93,654,153	2,133,835,051	3,469,306,894	-13,403,745,515	-9,934,438,621
5.0%	90.0%	90.0%	1.0%	5.0%	916	354	38.6%	113	31.9%	311	87.9%	2,398,125,878	-138,541,330	2,259,584,548	2,547,148,986	-11,201,617,255	-8,654,468,269
5.0%	90.0%	90.0%	2.0%	0.5%	916	366	40.0%	109	29.8%	360	98.4%	2,726,473,810	-4,051,621	2,722,422,189	3,184,222,709	-10,952,467,962	-7,768,245,253
5.0%	90.0%	90.0%	2.0%	1.0%	916	361	39.4%	117	32.4%	356	98.6%	3,077,381,392	-6,653,733	3,070,727,659	4,068,836,244	-12,320,920,107	-8,252,083,863
5.0%	90.0%	90.0%	2.0%	2.5%	916	366	40.0%	117	32.0%	342	93.4%	2,070,582,553	-47,680,136	2,022,902,417	3,450,297,359	-11,678,837,581	-8,228,540,222
5.0%	90.0%	90.0%	2.0%	5.0%	916	367	40.1%	123	33.5%	308	83.9%	2,596,820,271	-299,140,174	2,297,680,097	4,810,013,924	-14,355,740,383	-9,545,726,459

- 회사의 총 수익이 극대화하는 값은 위의 표와 같다. (납입 보험액: 5.0%, 수익/손해 환불 비율: 90% 고정)
- (상대적으로 비율이 낮지만) 약 30 ~ 35%의 고객이 해당 보험 가입으로 인해 이득을 봄
- (상대적으로 비율은 높지만) 회사 입장에서는 이득을 본 거래 수 중 88 ~ 97%를 통해 이득을 봄
- 회사의 수익 규모는 약 20 ~ 30억 정도

5. Simulation 결과 : 이득을 본 고객 비율 (상위 10)

납입 보험액	수익 환불 비율	손해 환불 비	수익 수수료	손해방지			전체 거래수	보험 신청수	보험 신청비율	이득 본 고객수	이득 본 고객비율	이득 본 회사 거래수	이득 본 회사 거래수 비율		회사 총 수익액	회사 총 손실액	수익/손실 금액(회사)		고객 총 수익액	고객 총 손실액	수익/손실 금액(고객)
0.5%	90.0%	90.0%	0.5%	0.5%			916	372	40.6%	174	46.8%	247	66.4%		257,584,848	-22,995,530	234,589,318		3,932,466,704	-12,251,979,116	-8,319,512,412
0.5%	90.0%	90.0%	0.5%	1.0%			916	359	39.2%	167	46.5%	227	63.2%		239,168,822	-64,252,560	174,916,262		3,578,174,768	-12,352,869,038	-8,774,694,270
0.5%	90.0%	90.0%	0.5%	2.5%			916	348	38.0%	160	46.0%	238	68.4%		180,696,330	-236,771,052	-56,074,722		3,967,360,244	-11,972,730,628	-8,005,370,384
0.5%	90.0%	90.0%	0.5%	5.0%			916	382	41.7%	175	45.8%	226	59.2%		153,585,489	-472,748,610	-319,163,121		4,358,026,299	-11,285,617,588	-6,927,591,289
0.5%	90.0%	90.0%	0.5%	10.0%			916	368	40.2%	167	45.4%	227	61.7%		141,458,342	-776,667,571	-635,209,229		4,008,678,572	-9,455,333,573	-5,446,655,001
0.5%	90.0%	90.0%	0.5%	20.0%			916	362	39.5%	164	45.3%	225	62.2%		121,290,108	-2,797,685,999	-2,676,395,891		3,510,756,191	-17,484,541,840	-13,973,785,649
0.5%	90.0%	90.0%	0.5%	50.0%			916	361	39.4%	163	45.2%	200	55.4%		159,760,615	-5,412,276,660	-5,252,516,045		4,403,627,462	-16,671,492,570	-12,267,865,108
0.5%	90.0%	90.0%	1.0%	0.5%			916	356	38.9%	161	45.2%	258	72.5%		194,974,496	-17,978,219	176,996,277		3,001,536,718	-8,645,860,797	-5,644,324,079
0.5%	90.0%	90.0%	1.0%	1.0%			916	374	40.8%	168	44.9%	222	59.4%		265,879,272	-44,065,048	221,814,224		3,201,536,456	-9,087,664,154	-5,886,127,698
0.5%	90.0%	90.0%	1.0%	2.5%			916	343	37.4%	154	44.9%	248	72.3%		245,037,983	-180,404,046	64,633,937		3,997,315,266	-9,984,631,721	-5,987,316,455

- 보험 가입으로 이득을 본 고객의 비율의 상위 값은 위의 표와 같다. (납입 보험액: 0.5%, 수익/손해 환불 비율: 90% 고정)
- 약 45%의 고객이 해당 보험 가입 이득을 보더라도, 회사 입장에서 수익이 날 수 있음을 확인
- (앞선 결과와 비교하며) 보험 거래 수 중 이득을 본 경우는 상대적으로 줄어들음, 약 65 ~ 72% 정도이다.
- 회사의 수익 규모는 약 1.7 ~ 2억 사이 (손해 발생 구간도 있음)

4

한계점 및

나아가야 할 방향

[한계점]

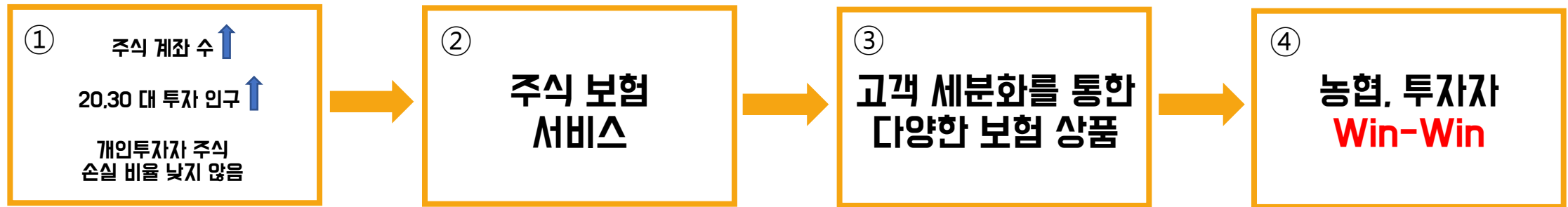
1. 시뮬레이션을 위해 판매금액 처리를 위한 가정들 존재
: 실제 판매 금액 데이터를 확보하여 계산 필요
2. 군집 결과에 대한 한계
: 군집화 진행 한정된 데이터로 클러스터를 진행
: 다양한 데이터를 통해 좀 더 수준 높은 클러스터 개발
: 해당 클러스터를 통한 보험 상품 개발

[나아가야 할 방향]

현재 보험 상품은 기존 고객들의 주식거래정보를 바탕으로 군집화하여 이를 기반으로 다양한 주식 보험 상품을 생성하였다. 하지만 신규 고객이 유입되었을 경우 해당 고객의 주식거래정보가 존재하지 않기 때문에 기존 고객들의 주식 거래 정보와 cus_info 데이터와 같이 고객의 인적 정보 또한 활용하여 주식 보험 상품을 추천하는 방식으로 해당 모델을 발전시키는 것을 제안한다.

5

결론



예금, 적금만으로 재산을
늘릴 수 없는 제로금리 시대
라는 시대적 배경으로 주식
투자자들이 점차 늘고 있음

현 사회의 변동성이 큰
주식시장에서 주식 투자자
들의 '손해'에 대한 근본적
인 두려움에 대한 접근으로
그들에게 심리적인 안정감
을 부여할 수 있음

농협의 수많은 데이터를
토대로 획일화된 보험 상품
이 아니라 고객의 특징에
따라 다양한 보험 상품을
만들어 그들의 needs를
충족시키는 보험 상품을
제안

주식 보험 서비스로 농협 측
은 신규 고객 확보 및 기존
고객 유지를 할 수 있고 기존
주식거래 수수료에 한정된 수
익구조에서 또 다른 수익이
발생할 수 있음

6

참고자료

6. 참고자료

주식 보험 서비스 구현 함수

```
def Profit_and_loss(df, ICR, PR, LR, PF, DP):
    company_pl_list = [] # 회사 이득 손해 0,1
    company_plc_list = [] # 회사 수익 손해 금액
    customer_pl_list = [] # 고객 이득 손해
    customer_plc_list = [] # 고객 순 수익/손해 금액
    for i in df.index:
        if df.iloc[i,-1] == 1:
            company_pl = 0
            cpl = (df.iloc[i,4] - df.iloc[i,5])*df.iloc[i,3] #고객 손익
            cht = int(df.iloc[i,2]/30)*1 #보유 기간
            IC = df.iloc[i,-2]*ICR*df.iloc[i,3] #납입 금액

            #company
            if (cpl >= 0):
                cop = cht*IC*(1-PR) #회사 납입 보험액 수익
                cf = int(cpl*PF) #고객 수입 수수료
                cpf = 0
            else:
                cop = cht*IC*(1-LR) #회사 납입 보험액 손해
                cpf = int(cpl*DP) #고객 보호 금액
                cf = 0

            company_pl = int(cop+cf+cpf) # 고객 수익/손해시 회사이득
            company_plc_list.append(company_pl) # 회사 이득 기록

            if (company_pl > 0):
                company_pl_list.append(1)
            else:
                company_pl_list.append(0)

            # customer
            cnp = int(cpl-cf-cop+cpf) # 고객 순 수익 손해
            customer_plc_list.append(cnp)

            if cnp >= cf+cop: # 고객 순 수익이 회사가 가져가는 보험료, 수수료 보다 클 경우
                customer_pl_list.append(1)
            elif (cpl < 0) and (cpf >= cop): # 고객이 손해 봤을 때 회사에 내는 수수료 보다 보장이 많이 볼때
                customer_pl_list.append(1)
            else:
                customer_pl_list.append(0)
        else:
            pass
    return company_pl_list, company_plc_list, customer_pl_list, customer_plc_list
```

군집 별 Simulation 자동화 코드

```
for num in [0.05, 0.1, 0.15, 0.2]:
    for j in [0.80, 0.85, 0.90, 0.925]:
        for m in [0.05, 0.1, 0.15, 0.2]:
            for d in [0.1, 0.2, 0.25, 0.5]:

                ### 각 부분 정의
                ICR_list.append(num)
                PR_list.append(j)
                LR_list.append(j)
                PF_list.append(m)
                DP_list.append(d)

                ### 새그먼트별 랜덤 주식 보험 신청 random화
                Pal_bas_df_0['보험신청'] = np.random.binomial(n=1, p=0.4, size=len(Pal_bas_df_0))
                Pal_bas_df_1['보험신청'] = np.random.binomial(n=1, p=0.4, size=len(Pal_bas_df_1))
                Pal_bas_df_2['보험신청'] = np.random.binomial(n=1, p=0.4, size=len(Pal_bas_df_2))
                Pal_bas_df_3['보험신청'] = np.random.binomial(n=1, p=0.4, size=len(Pal_bas_df_3))
                Pal_bas_df_4['보험신청'] = np.random.binomial(n=1, p=0.4, size=len(Pal_bas_df_4))
                Pal_bas_df_5['보험신청'] = np.random.binomial(n=1, p=0.4, size=len(Pal_bas_df_5))
                Pal_bas_df_6['보험신청'] = np.random.binomial(n=1, p=0.4, size=len(Pal_bas_df_6))

                company_pl_list, company_plc_list, customer_pl_list, customer_plc_list = Profit_and_loss(Pal_bas_df_1, num, j, j, m, d)

                ### 회사, 고객별 순수익 확인
                Pal_df = pd.DataFrame({'회사미산손익':company_pl_list,
                                      '회사손익금액':company_plc_list,
                                      '고객미산손익':customer_pl_list,
                                      '고객순손익':customer_plc_list})

                ac = 0
                ac1 = []
                for i in Pal_df.회사손익금액:
                    ac = ac+i
                    ac1.append(ac)
                Pal_df['회사손익누적금액'] = ac1

                # 정의하기

                total_transaction.append(Pal_bas_df_1.shape[0])
                insurance_cnt.append(len(Pal_df))
                insurance_ratio.append(round(len(Pal_df) / Pal_bas_df_1.shape[0] , 3))

                benefit_consumer_cnt.append(sum(Pal_df.고객미산손익))
                benefit_consumer_ratio.append(np.round(sum(Pal_df.고객미산손익)/len(Pal_df.고객미산손익), 3))

                benefit_company_cnt.append(sum(Pal_df.회사미산손익))
                benefit_company_ratio.append(np.round(sum(Pal_df.회사미산손익)/len(Pal_df.회사미산손익),3))

                company_benefit_total.append(sum(Pal_df.query('회사미산손익 == 1')['회사손익금액']))
                company_loss_total.append(sum(Pal_df.query('회사미산손익 == 0')['회사손익금액']))
                company_result.append(sum(Pal_df.query('회사미산손익 == 1')['회사손익금액'])
                                     + sum(Pal_df.query('회사미산손익 == 0')['회사손익금액']))

                consumer_benefit_total.append(sum(Pal_df.query('고객미산손익 == 1')['고객순손익']))
                consumer_loss_total.append(sum(Pal_df.query('고객미산손익 == 0')['고객순손익']))
                consumer_result.append(sum(Pal_df.query('고객미산손익 == 1')['고객순손익'])
                                     + sum(Pal_df.query('고객미산손익 == 0')['고객순손익']))
```

6. 참고자료

<https://www.mk.co.kr/news/society/view/2021/01/26577/> (매일경제, 김제림 기자)

https://biz.chosun.com/site/data/html_dir/2020/06/28/2020062800468.html (조선비즈, 박성우 기자)

<https://www.oreilly.com/library/view/statistics-for-machine/9781788295758/c71ea970-0f3c-4973-8d3a-b09a7a6553c1.xhtml> (OREILLY)

시뮬레이션 결과(군집 0.1.2) 결과 엑셀파일 별도 첨부

감사합니다