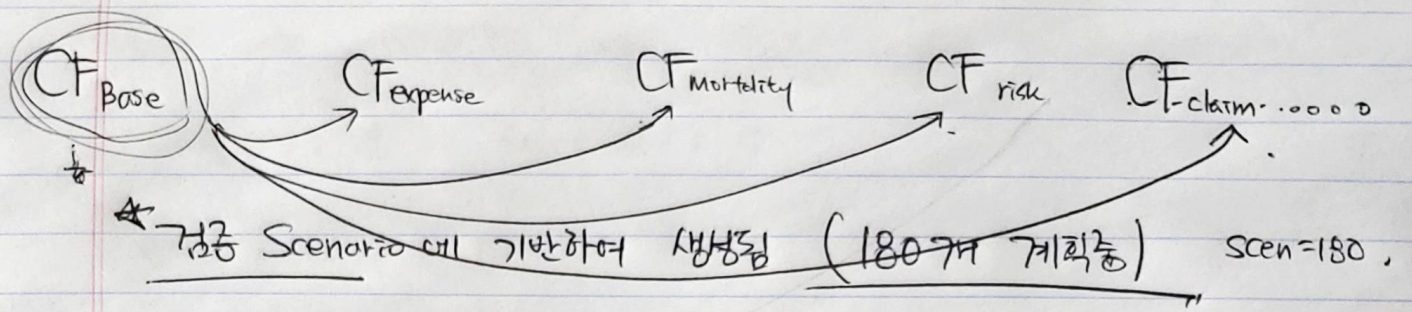
#1

Similarity Measure for CF matrix.

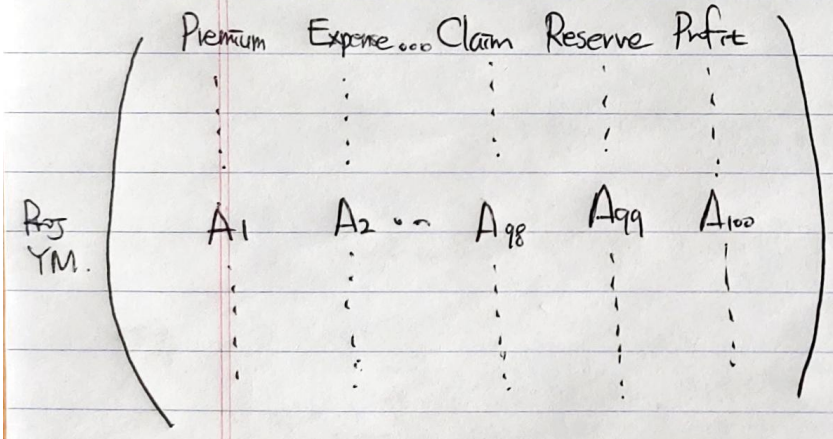
$CF_{matrix} = \mathbb{R}^{1200 \times 100}$ (변동요율 항목 100개 가정)



★. How to measure and discriminate difference.

i) Data format

CF matrix Base
(1200 x 100)



vector form.

$A_1 = (a_{11}\ a_{21}\ \dots\ a_{1200\,1})$

$A_2 = (a_{12}\ a_{22}\ \dots\ a_{1200\,2})$

$A_{100} = (a_{1\,100}\ a_{2\,100}\ \dots\ a_{1200\,100})$.

↳ Same for $CF_{expense}$ $CF_{Mortality}$ CF_{risk} CF_{claim}.

#2

ii) Similarity Measure for vector

↳ to measure direction of vector (up & down).

we use property of cosine. & inner product.

$\cos(0°) = 1$ $\cos(90°) = 0$.

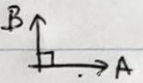
$A = (a_1, a_2 \dots a_n)$

$B = (b_1, b_2 \dots b_n)$.

$A \cdot B = \|A\| \times \|B\| \times \cos\theta$.

where $\|A\| = \sqrt{a_1^2 + a_2^2 + \dots + a_n^2}$, $\|B\| = \sqrt{b_1^2 + b_2^2 + \dots + b_n^2}$ (L2 norm

example $A = (1, 0)$. $B = (0, 1)$.



$A \cdot B = 1 \times 1 \times 0 = 0$! → ⓐ Zero Similarity.

⇒ Similarity_measure.py 모듈내

cos_similarity( ) 함수로 구현되어 있음.

$$= \frac{\sum_{i=1}^{100} a_{1i}\, b_{1i} \cos\theta}{\sqrt{\sum_{i=1}^{100} a_{1i}^2} \cdot \sqrt{\sum_{i=1}^{100} b_{1i}^2}} = \bigcirc\% = \text{scalar}.$$

↳ L2 norm

#3

III) Similarity (Matrix (Affinity Matrix). ~~from similar~~
(Vector

(II) 에서 산출한 CF_vector 간 유사도를 Vector (~~Matrix~~) 형태로 저장.

(Python 에서는 Matrix 자료형을 Pandas DataFrame 에 저장.
(Python 에서는 Vector 자료형은 List 에 저장.

(CF_{Base} Matrix) 와 (CF_{claim} Matrix) 의 Similarity Vector로 (~~Matrix~~).

$$CF_{Base} = \begin{pmatrix} Premium & Expense & Claim & Profit \\ \vdots & \vdots & \vdots & \vdots \\ A_1 & A_2 & A_3 & A_4 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}, \quad CF_{claim} = \begin{pmatrix} Premium & Expense & Claim & Profit \\ \vdots & \vdots & \vdots & \vdots \\ B_1 & B_2 & B_3 & B_4 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

Similarity (CF_{Base}, CF_{claim}) = Vector(List).

= [Premium similarity = sim(A_1 . B_1) , Expense Similarity = sim(A_2 . B_2) , ... , Claim Similarity = sim(A_3 B_3) , Profit Similarity = sim(A_4 . B_4)]

↓ Scalar

= example [100% , 100% , ... , 98% , 99%].

⇓

< Similarity Matrix form > ,

	Premium	Expen.	Claim	Profit →항목
CF Base	100%	100%	100%	100%
CF expense	99%	95%	100%	98%
CF claim	99%	100%	94%	93%
CF Mortality	100%	100%	96%	96%
CF Risk	100%	100%	96%	98%

Base 에 대비하여 ⇒

⇓ ~~매트릭스~~ 시나리오개수 = 180개.

#4 (IV)

검증 Scenario 180 개를 준비하였음.

(output_scenario.xlsx 파일 참조).

CF Base 를 기반으로 한 Similarity Matrix (III) 참조.

(180 x 100) Matrix 가 나옴 = SC Base Matrix

180 = #Scenario. 100 = # CF items.

⇓

분류를 위해서는 단순합산 Score 로는 부족함.

(V) Anormaly Detection With Graph Theory.

⟨Guilty_By_Association⟩ technique.

↳ Malicious web page detection에 사용되는 기술.

Blue Node : CF item = 100 개

Red Node : Scenario = 180 개.

Edge : Link Blue Node & Red Node

if (Similarity ≠ 1) = different.

(example) ~~Scenario expense 에서~~와 CF expense 와 CF Base

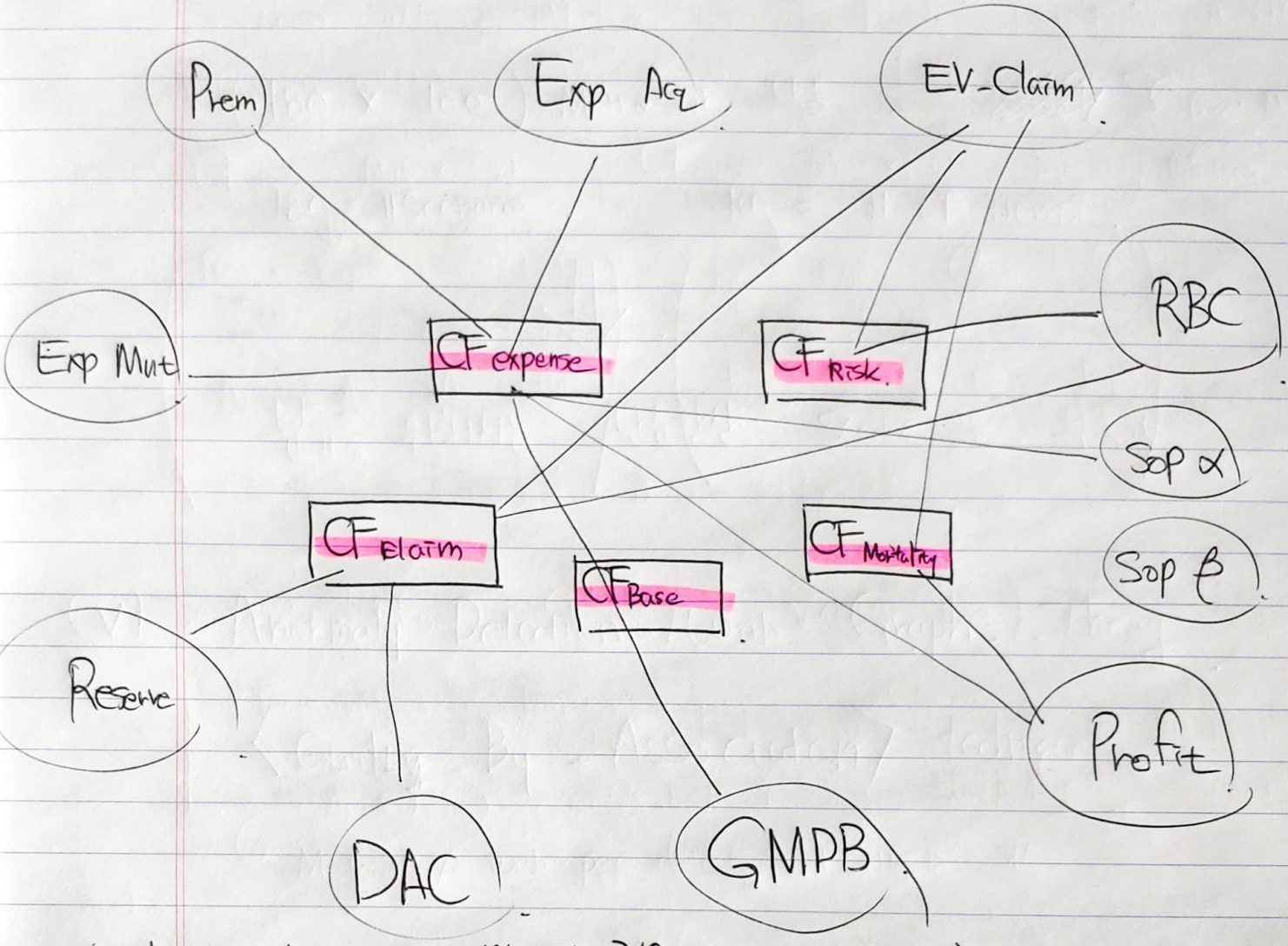
의 경우에 Exp Acq 현금흐름 항목이 유사하지 않음

⇒ CF exp 와 Exp Acq 를 연결하는 Edge.

#5

< CF Base 를 기준으로 한 Similarity Matrix ⇒ Graph >.

Blue Node = 100개 현금흐름 항목. Red Node = 180 개 시나리오 현금흐름 Matrix.



(abnormality 가 발견된 경우에만 graph 연결)

* network X 'library 를 사용하여 구현함

graph. py 파일 참조 바람.

#6

(VI) How to measure Similarity among Red Node.

(180 74 시나리오 CF matrix기준의 유사도).

We built directed graph G.

and u, v represent node among G.

we are going to measure red node (V) only.

I(v) = set of in-neighbors node of node V.

O(v) = set of out-neighbors node of node V.

$1 \le i \le |I(v)|$, $1 \le i \le |O(v)|$

graph similarity between red node a, and red node b.

$= S(a,b) \in [0,1]$.

$0 \le S(a,b) \le 1$, $S(a,b) = S(b,a)$

$S(a,a) = S(b,b) = 1$.

$$S(a,b) = \frac{C}{|I(a)| \cdot |I(b)|} \sum_{i=1}^{|I(a)|} \sum_{J=1}^{|I(b)|} S(I_i(a), I_j(b))$$

(where C is ~~Cons~~ importance factor constance

in here we used $C = 0.9$.)

↳ Degree Base Approach.

#7

(VII) Final output of Similarity Score.

('CF Base 를 기준으로 하여...)

Scen #	Base	#1	#2	#3	...	#180
Base	100%	1%	2%	6%		0.1%
#1		100%	1%	3%		0.2%
#2			100%	2%		0.1%
#3				100%		0.5%
⋮						
#180						100%

↳ In here % score means how many abnormal CF items their scenarios share.

⇒ Do you have a guilty friend around you?

각 Row 별로 Rank 를 세워서 유사성 추적 가능.