

## ##33. zenspark\_noweight\_bootstrap\_feature7.py 모델 분석 및 튜닝 Q&A 전체 기록

### Q: ##33. zenspark\_noweight\_bootstrap\_feature7.py 에서 그리드 서치 대상 하이퍼파라미터라는데 그 의미를 간단하게 설명

A: 이 파일에서 "그리드 서치 대상 하이퍼파라미터"는 모델 성능을 비교하려고 여러 값 조합으로 바꿔가며 시험하는 설정값이라는 뜻입니다.

### 33. zenspark\_noweight\_bootstrap\_feature7.py 기준으로는:

- **use\_sample\_weight**: 샘플 가중치 사용 여부 (False/True)
- **l1\_ratio**: ElasticNet에서 L1/L2 규제 비율
- **C**: 규제 강도의 역수(클수록 규제가 약해짐)
- **min\_recall**: 임계값 선택 시 최소 재현율 조건

즉, 위 값들의 조합( $2 \times 5 \times 7 \times 3 = 210$ 개)을 돌려서 CV/백테스트 기준으로 가장 좋은 조합을 고르는 과정입니다. 참고로 THRESHOLD\_GRID는 확률을 0/1로 자르는 분류 임계값 후보를 찾는 용도입니다.

### Q: MAX ITER = 5000 인이유

A: MAX\_ITER = 5000인 이유는 `LogisticRegression(solver="saga", penalty="elasticnet")` 가 수렴에 시간이 더 걸릴 수 있어서입니다. 핵심은 3가지입니다.

1. saga + elasticnet 조합은 반복 횟수가 부족하면 수렴 실패가 자주 납니다.
2. 이 스크립트는 하이퍼파라미터 조합/부트스트랩을 많이 돌리므로, 일부 조합에서만 수렴 실패해도 비교가 불안정해집니다.
3. 그래서 반복 상한을 크게 잡아 수렴 안정성을 우선한 설정입니다(대신 계산시간은 늘어남).

즉, 성능을 올리기 위한 값이라기보다, 학습이 끝까지 안정적으로 수렴하도록 하는 안전장치에 가깝습니다.

### Q: 지금 모델에서 다중공선성 경고 되는게 있나? 관련된 코드를 넣긴했는데

A: 현재 기준으로는 다중공선성 경고 안 뜹니다.

**코드 로직:**  $|corr| \geq 0.92$  이  $|corr| \geq 0.92$ 인 feature pair가 있으면 경고 출력

제가 현재 데이터(01. TEST\_KJC.csv)로 동일 조건 계산해본 결과:

**high\_corr\_count = 0** (임계값 이상 pair 없음)

즉, 경고 코드 자체는 정상으로 들어가 있고, 지금 데이터에서는 조건을 만족하는 쌍이 없는 상태입니다.

### Q: 그리고 extra feature 안정성 기준이 들어가있는데 Lasso가 60%이상 경우에서 0이 아니어야 하는 것과 CV가 2.5이하여야 pass 인 이유

A: 이 기준은 이론적 정답이라기보다, "추가 feature가 진짜 신호인지"를 걸러내는 운영용 휴리스틱입니다.

- **nonzero\_rate  $\geq 0.60$ :** L1(라쏘 성분)이 있는 ElasticNet에서 계수가 0이 자주 되면, 그 feature는 샘플 재추출(bootstrap)마다 선택이 흔들린다는 뜻입니다. 60% 이상은 "적어도 과반을 넘는 다수 반복에서 살아남는가"를 보는 최소 안정성 컷입니다.
- **coef\_cv  $\leq 2.5$ :** 여기서  $coef\_cv = \frac{coef\_std}{coef\_mean}$  이라서, 계수 변동성이 평균 크기 대비 너무 크면 불안정하다고 봅니다.  $2.5 \leq 2.5$ 는 "표준편차가 평균 절대값의 2.5배 이내"라는 뜻으로, 방향/크기 흔들림이 과한 feature를 제외하려는 컷입니다.

즉, 두 조건을 같이 쓰는 이유는:

1. **선택 안정성**(자주 0 되지 않는가) +
2. **크기 안정성**(살아남아도 계수가 너무 출렁이지 않는가)  
를 동시에 보려는 것입니다.

---

## Q: 리스트 제어 정책이 있는 이유

**A:** 여기서 말한 “리스트 제어 정책”은 코드상 Risk-control policy (fixed rules)와 같은 맥락으로 보입니다. 즉, 하이퍼파라미터 후보 “리스트”를 고를 때 우선순위를 고정해 과적합/임의선택을 막는 장치입니다.

이 정책이 필요한 이유:

1. 성능지표가 여러 개라서, 기준 없으면 그때그때 유리한 지표만 보고 모델이 바뀌기 쉽습니다.
2. 연도별 변동이 큰 데이터라 단일 지표 최적화는 재현성이 떨어집니다.
3. 운영 관점에서 중요한 목표(Top-k hit, backtest 안정성)를 먼저 보장하려면 선택 순서를 고정해야 합니다.

결과적으로, “최고 점수 1개”보다 일관되고 설명 가능한 모델 선택을 보장하려고 넣은 정책입니다.

---

## Q: SHAP 의존도 플롯에 표시할 feature 가 왜 3이야?

**A:** 코드 기준으로는 `TOP_DEPENDENCE_FEATURES = 3`가 정책값(표시 개수 제한)으로 하드코딩돼 있고, 아티팩트에 넘겨서 플롯 스크립트가 그 개수만 그리게 되어 있습니다.

왜 3이냐를 실무적으로 풀면:

1. feature가 7개라 의존도 플롯을 전부 그리면 해석 포인트가 분산됨
2. 상위 3개만 보면 “주요 관계”를 빠르게 확인 가능
3. 보고서/슬라이드에 들어갈 시각화 밀도(가독성) 관리 목적

즉, 통계적으로 반드시 3이어야 해서가 아니라, 해석성과 출력물 길이 제어를 위한 기본값입니다. 원하면 3 -> 5처럼 바로 늘릴 수 있습니다.

---

## Q: pickle이 뭐임

**A:** pickle은 Python 객체를 파일로 저장/복원하는 직렬화 모듈입니다.

- **저장:** 메모리의 모델, 딕셔너리, 리스트 등을 바이트로 변환해 `.pk1`에 기록
- **복원:** 파일에서 다시 읽어 원래 Python 객체로 되살림

이 코드에선 학습 결과(`final_model`, `features` 등)를 `model_artifacts.pk1`로 저장해서, 나중에 플롯 스크립트가 그대로 재사용하게 하려는 용도입니다.  
(주의: pickle 파일은 신뢰 가능한 출처만 로드해야 합니다(임의 코드 실행 위험).)

---

## Q: 플롯 스크립트는 뭔데

**A:** 플롯 스크립트는 말 그대로 학습은 안 하고, 저장된 결과(`.pk1` 등)로 그래프만 그리는 분리된 스크립트입니다.

이 파일에서는:

1. 모델 학습 후 `model_artifacts.pk1` 저장
2. 이후 “플롯 스크립트를 따로 실행”하라고 안내

즉 목적은 **학습/탐색 코드와 시각화 코드를 분리하고 재학습 없이 빠르게 그림만 다시 뽑는 것**입니다. 참고로 지금 폴더 기준으로는 안내문에 나온 `##33. zenspark_noweight_weighted_bootstrap_novelty_removed_plot_only.py` 파일이 실제로는 보이지 않아서, 파일명이 바뀌었거나 누락된 상태로 보입니다.

## Q: saga가 뭐야

A: 아마 saga는 **saga** 오타로 보입니다. SAGA는 scikit-learn LogisticRegression의 최적화 알고리즘(솔버)입니다.

- L1/L2/ElasticNet 규제를 지원
  - 희소 해(계수 0 만들기)에 강함
  - 대용량/희소 데이터에도 잘 쓰임
- 이 코드에서 `penalty="elasticnet"` 을 쓰기 때문에 `solver="saga"` 를 선택한 겁니다.

## Q: 임계값 적용해 이진 예측 생성이 무슨말

A: 뜻은 간단히 이겁니다.

1. 모델이 먼저 확률값을 냅니다. 예: 0.73, 0.21
2. 정한 임계값(threshold)과 비교해서 0/1로 바꿉니다.
3. 보통 규칙: 확률  $\geq$  임계값이면 1, 아니면 0

즉 “임계값 적용해 이진 예측 생성”은 **확률 예측을 최종 클래스(양성/음성)로 변환하는 단계**라는 뜻입니다.

## Q: 샘플수를 기반으로 학습 가중치 생성한다는 말?

A: 각 학습 샘플에 같은 비중을 주지 않고, n(샘플수) 컬럼을 이용해 **\*\*가중치(sample\_weight)\*\***를 만들어 학습에 반영한다는 뜻입니다.

이 코드에서는 대략:

- n이 클수록 가중치가 커짐 ( $\sqrt{n}$  기반)
- 중앙값으로 정규화
- 너무 크거나 작은 값은 0.5~2.0으로 클리핑

즉, 표본이 많은 클레이드-연도 행의 신호를 조금 더 신뢰하되, 한쪽이 과도하게 지배하지 않게 제한하는 방식입니다.

## Q: bootstrap 이란게 뭐야 / bootstrap 계수 안정성?

A: **bootstrap**은 데이터를 여러 번 “복원추출”해서 다시 학습해보는 방법입니다. **bootstrap 계수 안정성**은:

1. 매번 대표본으로 모델을 다시 학습하고
2. 각 feature 계수(회귀계수)를 모은 뒤
3. 얼마나 일관적인지 보는 것

이 코드에선 대표적으로 `nonzero_rate` (계수가 0이 아닌 비율), `coef_std` (계수 변동성/표준편차), `coef_cv = coef_std / |coef_mean|` (평균 대비 흔들림)을 써서, feature가 우연히 잡힌 건지(불안정) 비교적 안정적 신호인지 판단합니다.

## Q: 상삼각 행렬만 탐색한다는게 뭐야?

A: 상관행렬에서 중복 비교를 피하려고 대각선 위쪽(상삼각)만 본다는 뜻입니다.

이유:

- `corr(A,B)` 와 `corr(B,A)` 는 같은 값이라 둘 다 볼 필요가 없음
  - `corr(A,A)=1` 인 대각선도 의미 없음
- 그래서 (i,j)에서  $j > i$  인 쌍만 검사하면, 모든 feature pair를 한 번씩만 효율적으로 확인할 수 있습니다.

**Q: 근데 우리 test+val clade 분석에서 B 다음으로 K가 많은데 그리고 validation 그래프 k가 많은데 이미 기존에 k가 많아서 2026년을 k로 예측한것은 아닌가?**

A: 네, 그 의심이 타당합니다. 현재 설정이면 K가 많이 나오는 구조 영향이 큼니다.

핵심 근거:

1. 로그에서 2025 K 비중이 48.2% (604/1252)이고, 2026 후보 #1도 K입니다.
2. 모델 feature에 freq가 직접 들어갑니다.
3. 백테스트는 val\_year의 clade들을 점수화해서, 같은 val\_year의 dominant와 비교합니다.

제가 같은 백테스트 연도(14개)에 대해 “모델 없이 freq 최대 clade 선택” baseline을 계산해보니 naive Top-1 = 14/14, naive Top-3 = 14/14가 나왔습니다. 즉 현재 프레임에서는 “이미 많은 clade를 따라가는 효과”가 매우 강합니다. 질문하신 대로, 2026을 K로 본 이유에 기존 prevalence가 크게 작용했을 가능성이 높습니다.

**Q: 그럼 우리가 진짜 내년에 새로 뜰 신호를 잡았는데 그게 k라고 보고싶다면 어떤식으로 접근해야해**

A: K를 “진짜 선행 신호”로 주장하려면, 아래 4가지를 충족해야 합니다.

1. **평가 설계부터 엄격하게 고정:** 반드시 t년 feature → t+1년 dominant만 평가. 같은 해(t)의 빈도 순위와 비교하는 평가는 보조로만 사용.
2. **강한 baseline을 이겨야 함:** 최소 비교인 naive\_freq(올해 점유율 1등), persistence(작년 dominant=올해 dominant)보다 모델이 반복적으로 좋아야 “추가 신호”라고 말할 수 있음.
3. **누수/동시성 제거된 feature만 사용:** freq(동시점 점유율) 대신 lag feature 위주(freq\_prev, freq\_delta, freq\_delta\_2y, 변이 증가율의 lag)를 사용. “2026 예측”이면 2025 말 기준으로 실제 이용 가능한 값만 포함.
4. **설명 + 안정성까지 확인:** K가 1등인 이유를 SHAP/계수로 제시(어떤 lag 신호가 K를 밀어올렸는지)하고, 부트스트랩/연도 서브샘플에서 K 1위가 얼마나 유지되는지(재현률) 함께 제시.

**Q: 그래서 결과가 뭐야? (Strict t->t+1 결과)**

A: 핵심 결과만 정리하면:

- **기존 백테스트:** Top-1 14/14 (78.6%), Top-3 14/14 (92.9%)
- **새로 추가한 엄격 백테스트(t -> t+1):** Top-1 13 (23.1%), Top-3 13 (53.8%)
- **2026 후보(2025 VAL 기준):** 여전히 K가 1순위 (예측 CR: 1/9 clades)

해석: 기존 지표는 높게 나오지만, 시간 누수 없는 엄격 기준에선 성능이 크게 낮아져서, “진짜 선행 예측력”은 아직 약한 상태로 봐야 합니다. 모델이 “내년에 새로 뜰 신호”를 잡았다기보다, “올해 이미 많은 클레이드”를 그대로 1등으로 뽑는 방식과 결과가 거의 같다는 뜻입니다.

**Q: 일단 여기까지의 위 결과 의미를 자세히 해석해줘 (K-event 포함)**

- A: 1. **기존 Backtest가 높은 이유:** 이 평가는 구조상 “그 해의 후보들”을 맞추는 성격이 강해서, 실제 선행 예측력보다 높게 나올 수 있습니다.
2. **Strict t->t+1가 진짜 핵심 지표:** t년 정보로 t+1년 dominant를 맞추면 성능이 낮습니다(23.1%). 즉, 미래 선행 신호를 강하게 잡고 있다고 보기 어렵습니다.
3. **Baseline과 동률이라는 의미:** strict 기준에서 모델(3/13, 7/13)과 naive freq(3/13, 7/13)가 같습니다. 즉 현재 모델의 추가 feature/학습 로직이 baseline 대비 “추가 예측력”을 보여주지 못했습니다.
4. **K-event 진단의 의미:** ActualK=1(정답이 K인 해 1번), PredK=0(모델이 K를 1등으로 예측한 적 없음). 즉, K를 맞춘 실적이 없어 “K를 미리 포착했다”는 근거가 없습니다.

**Q: 아니 그러면 모델이 선행 신호로 k를 검출했다라는 건 없는거야? 아예? 아니 그러면 내가 모델 튜닝은 왜 했어. 그랬음에도 불구하고 처음에 모델이 성능이 아주 엉망진창이었잖아**

A: 지금 결과 기준으로는, 맞습니다. “K를 선행 신호로 검출했다”는 주장을 현재 평가지표로는 뒷받침하기 어렵습니다. 하지만 이게 “튜닝이 무의미했다”는 뜻은 아닙니다.

1. **튜닝으로 개선된 건 사실:** 초기 엉망이던 모델을 최소한 작동 가능한 상태로 만들었고, 기존 운영형 지표(backtest/CV)는 크게 좋아졌습니다.
2. **개선의 성격 문제:** 그 개선이 “진짜 미래 예측력” 개선인지, 아니면 “현재 분포/순위 맞추기” 개선인지가 strict 평가에서 걸린 겁니다.

3. **결론:** 튜닝은 파이프라인 안정화/성능 회복 단계를 끝낸 것이고, 지금부터가 “선행성 검증” 단계입니다. 현재는 K-emergence 양성(next dominant = K) 사건이 데이터상 1건뿐이라 선행 검출을 학습/검정하기 어려운 한계가 있습니다.

**Q: ##33. zenspark\_noweight\_bootstrap\_feature7.py 에서 M33 SHAP 이랑 final model coefficients를 왜 이런 feature들을 선택되었는지 설명이 필요해 (AUC, PR-AUC, recall 항상 목적 포함)**

**A:** M33에서 저 feature들이 “자동으로 뽑힌 이유”는 **성능 목표 + 시간누수 방지 + 안정성 검증 + ElasticNet 계수수축**이 같이 작동한 결과입니다.

1. **정확성보다 예방 목적:** 감염병 특성상 미탐지를 줄이기 위해 \*\*재현율(Recall)을 먼저 고정(min\_recall)\*\*하고, 그 범위 안에서 AUC와 PR-AUC를 최적화했습니다.
2. **안정적 피쳐 선정:** 임계값 탐색과 ElasticNet 규제를 통해 일시적 노이즈보다 여러 시점에서 일관되게 신호를 주는 변수들만 안정적으로 남았습니다.
3. **최종 결과:** 그 결과 장기 추세( freq\_delta\_2y )와 최근 증가율( freq\_delta , rolling\_median\_freq )이 가장 안정적으로 살아남아 핵심 피쳐로 선정되었습니다.

**Q: 우리 7개 피쳐 의미와 뜻 설명 (간략하게)**

- A:**
1. **n:** 해당 연도-클레이드의 시퀀스 수(관측 표본 수).
  2. **freq:** 해당 연도 전체 대비 그 클레이드 비율(상대 빈도).
  3. **freq\_delta:** 전년 대비 빈도 변화량 ( $freq_t - freq_{t-1}$  freqt - freqt-1).
  4. **freq\_delta\_2y:** 2년 전 대비 빈도 변화량 ( $freq_t - freq_{t-2}$  freqt - freqt-2).
  5. **rolling\_median\_freq:** 최근 3년 빈도의 중앙값 (단기 노이즈 제거 추세 지표).
  6. **nonsyn\_med:** 해당 클레이드의 비동의 아미노산 치환 수 중앙값 (단백질 변화 강도 지표).
  7. **pam\_reversion\_med:** 원래 상태로 되돌아간(reversion) 아미노산 치환 수 중앙값 (역변이 경향 지표).

**Q: 근데 우리 n이 효과가 없지만 남겨둔 이유**

**A:** n을 남겨둔 이유는 “강한 예측 신호”라서가 아니라, **안전장치 역할** 때문입니다.

1. n은 해당 clade-year의 표본 신뢰도를 보여주는 기본 정보입니다.
2. 실제 학습 결과에서 계수가 0에 가깝게 수축했다는 건, 모델이 “필요 없으면 거의 안 쓰도록” 자동 조절했다는 뜻입니다.
3. 남겨둬도 성능을 크게 해치지 않고(ElasticNet이 억제), 데이터 분포가 바뀌는 미래 시점에는 보조 신호로 작동할 가능성을 열어둔 것입니다.

끝.