Oscillator scoring

+ 생존자 편향, 네이버 댓글 키워드

01.28

Oscillator

: 해석

$$\text{Oscillator(t)} = 10\%\ \text{trend(t)} - 5\%\ \text{trend(t)}$$

$$s.t.$$

$$10\%\ trend(t) = 0.90 \times 10\%\ trend(t-1) + 0.10 \times (\text{AV(t)} - \text{DV(t)})$$

$$5\%\ trend(t) = 0.95 \times 5\%\ trend(t-1) + 0.05 \times (\text{AV(t)} - \text{DV(t)})$$

AV(t): t 시점 상승한 종목들의 총 거래량
DV(t): t 시점 하락한 종목들의 총 거래량

Sub Index3(Stock Price Breadth)은 McClellan 거래량 합계 지수를 지표로 사용하는데 이 지수는 시장에서 상승 및 하락하는 주식의 누적 수를 측정하여 거래량의 비율을 계산한다. McClellan 거래량 합계 지수가 상승하면 시장의 전반적인 추세가 상승세이므로 탐욕의 신호로 판단, 하락하는 경우 추세는 하락세이므로 공포 신호로 판단한다[25].

- Oscillator가 **양수** → 상승 종목 거래량이 우세 → 시장 강세
- Oscillator가 **음수** → 하락 종목 거래량이 우세 → 시장 약세

- (오른쪽 내용) : 공포-탐욕 지수를 이용한 주식 수익률 예측모형 연구 -> 논문 발췌 내용

생존자 편향 (Survivorship Bias)

1. 정의

: 주로 성공 사례만 분석할 때 발생
실패 사례를 무시해 왜곡된 패턴을 학습

2. 원리

: 모델 학습 시 성공한 기업만 이용하여 학습하게 되면 그 전략이 보편적이라고 오인.
→ 실패한 경우를 학습하지 않아 낙관적 예측이 될 가능성이 높아짐.

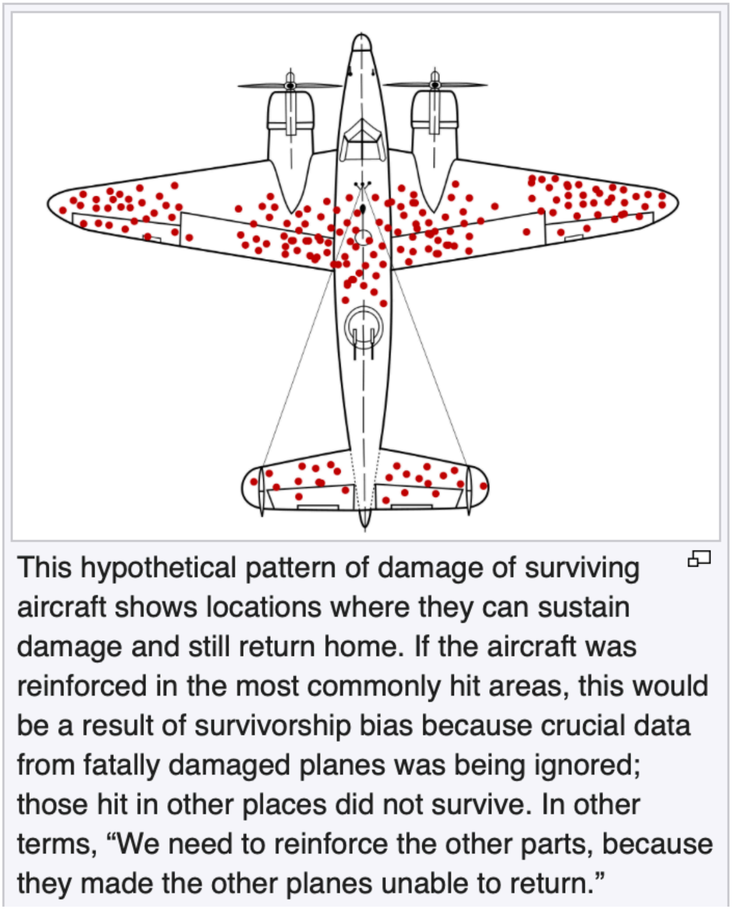
3. 현재 프로젝트에 대입

: 주가 폭은 시장 전체 강 혹은 약을 보는 지표
→ 나중에 망해서 사라진 지표들도 분명 그 당시 시장의 일부

만약 상장폐지된 종목을 빼버리면 끝까지 살아남은 강한 종목들을 위주로 계산
→ 시장 변동성이 덜 반영됨

4. 실무

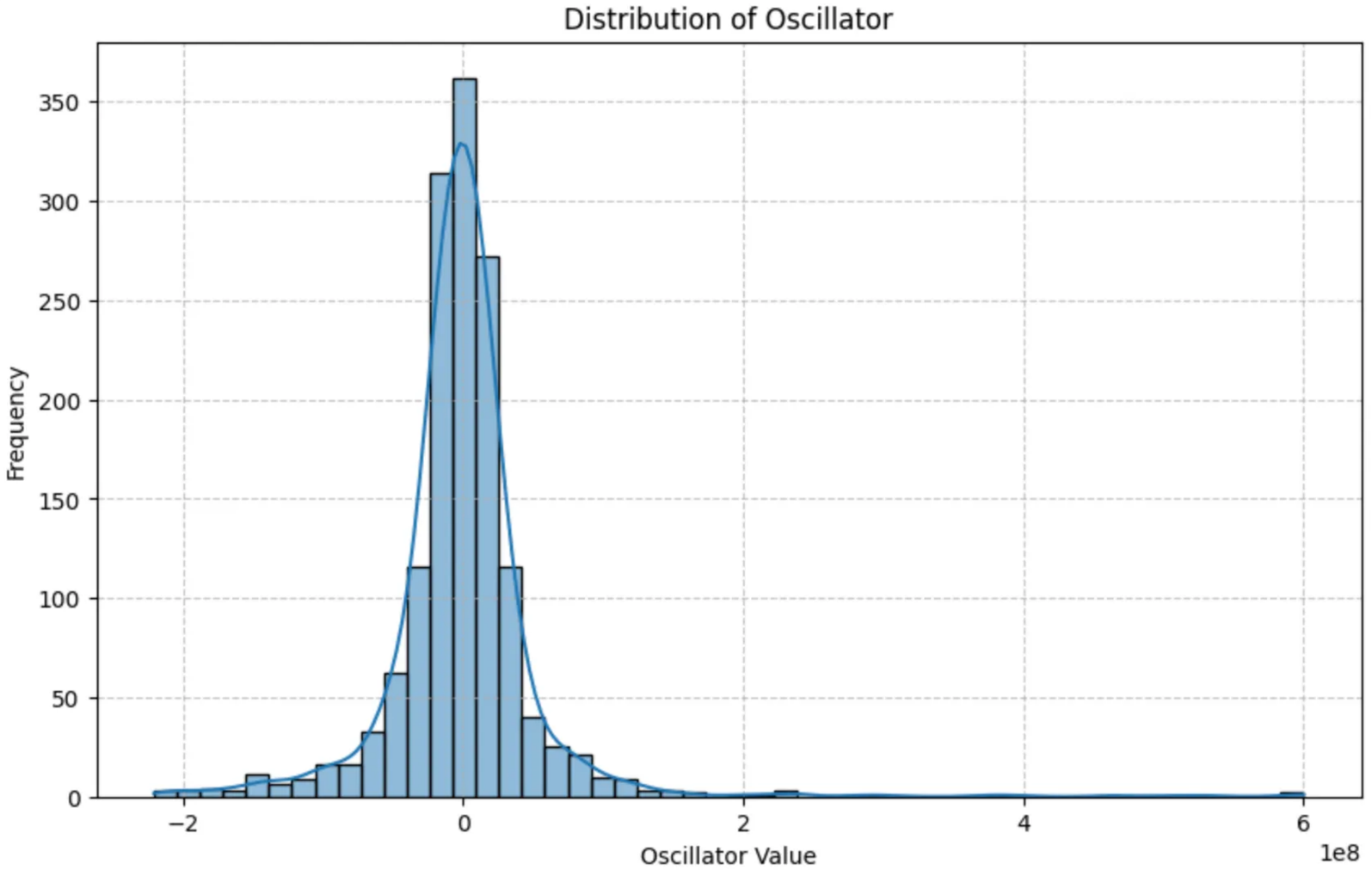
backtest : 과거 특정 시점 t에서 관측 가능했던 종목 집합 전체 기준으로 다 포함시킴



This hypothetical pattern of damage of surviving aircraft shows locations where they can sustain damage and still return home. If the aircraft was reinforced in the most commonly hit areas, this would be a result of survivorship bias because crucial data from fatally damaged planes was being ignored; those hit in other places did not survive. In other terms, “We need to reinforce the other parts, because they made the other planes unable to return.”

Oscillator scoring

전체 분포 확인



: 0을 중심으로 대칭인 형태

오른쪽으로 꼬리가 조금 긴 편

(최댓값이 과하게 큼)

```
1~99% 밖 비율: 2.0942%
0.5~99.5% 밖 비율: 1.0471%
```

Oscillator scoring

점수화 후보

1. Strength score 방식

장점 : 매우 직관적
실제 레포트 방향과도 잘 맞음

단점 : 분포가 조금 비대칭이면 위험함.
oscillator의 경우 outlier가 다수 존재 → outlier에 민감함

2. Rolling 정규화 + CDF값*100

장점 : outlier에 둔감, 분포가 왜곡되어도 비교적 안정적
fear&greed index계산시에도 해석 동일

단점 : 최고/최저에 대한 절대적 기준은 약화됨

* 결론

: 다수의 이상치가 존재하고, 여기에 지배당할 위험이 주가폭은 크다고 판단

→ **Rolling(1년을 window로 사용) 정규화 이용**

Oscillator scoring

Scoring flow

1. 시점별 정규화 (Rolling Z-score)

$$\mu_t = \frac{1}{N}\sum_{i=0}^{N-1} Osc_{t-i}, \quad \sigma_t = \sqrt{\frac{1}{N-1}\sum_{i=0}^{N-1}(Osc_{t-i} - \mu_t)^2}, \quad where \quad N = 252$$

$$Z_t = \frac{Osc_t - \mu_t}{\sigma_t}$$

2. Z-score clipping (이상치 방지)

$$\tilde{Z}_t = \min\left(\max(Z_t, -c), c\right), \quad where \quad c = 5$$

3. 0~100 점수화

$$\text{Score}_t = 100 \cdot \Phi(\tilde{Z}_t)$$

Oscillator scoring

데이터 정제 과정 요약

1.데이터 수집 (KOSPI)

1.2018년부터 raw data수집
2.EMA 기반 oscillator 계산에 사용

kospi_2018_2025_ohlcv

	날짜	ticker	시가	고가	저가	종가	거래량	등락률
0	2018-01-02	5930	51380	51400	50780	51020	169485	0.11773940345368900
1	2018-01-03	5930	52540	52560	51420	51620	200270	1.1760094080752600
2	2018-01-04	5930	52120	52180	50640	51080	233909	-1.0461061604029400
3	2018-01-05	5930	51300	52120	51200	52120	189623	2.0360219263899800
4	2018-01-08	5930	52400	52520	51500	52020	167673	-0.19186492709132800

2.지표 생성

1.2019년부터 McClellan Oscillator 생성
2.EMA(0.10, 0.05)는 과거 누적 영향이 큼
3.2018년을 EMA 워밍업 구간으로 사용

oscillator_2019_2025

date	oscillator
2019-01-02	-2376621.4861776200
2019-01-03	155959.83530869300

3.점수화 (normalization)

1. 최근 1년(252 거래일 기준) rolling mean/std 계산
2. Rolling z-score 계산
3. Z-score clipping 이후 표준 정규분포 CDF -> [0,1]
4. 위 점수에 100 곱하기

osc_score_2020_2025

date	osc_score_0_100
2020-01-02	93.29910231025150
2020-01-03	96.07998388194770

Oscillator scoring

Python code

```python
from scipy.stats import norm

out = out.copy()
out["date"] = pd.to_datetime(out["date"])
out = out.sort_values("date")

window = 252          # 최근 1년
clip_c = 5            # 점수용 클립(원하면 None으로 끄기)

# rolling mean/std
roll = out["oscillator"].rolling(window, min_periods=window)
mu = roll.mean()
sd = roll.std(ddof=1)

# rolling z-score
out["osc_z_1y"] = (out["oscillator"] - mu) / sd

# 점수용 z
z_for_score = out["osc_z_1y"]
if clip_c is not None:
    z_for_score = z_for_score.clip(-clip_c, clip_c)

# 0~100 점수 (표준정규 CDF * 100)
out["osc_score_0_100"] = norm.cdf(z_for_score) * 100

# 저장 (NaN 제거: 처음 252일은 rolling 때문에 NaN)
score_df = out.loc[out["osc_score_0_100"].notna(), ["date", "osc_score_0_100"]].copy()
score_df.to_csv("osc_score_2020_2025.csv", index=False, encoding="utf-8-sig")
```

: out -> oscillator 날짜별 저장된 Dataframe

네이버 댓글 키워드

네이버 댓글 감성분석을 활용하는 것 : 특정 종목에 국한 되는 것 보단 시장 전체 심리 파악하는 데 강점 존재
→ 사실 잘 모르겠음

1) 시장 지수에 직접적인 키워드

코스피, 코스닥, 국내증시, 증시, 주식시장, 한국증시

2) fear or greed를 직접적으로 파악할 수 있는 변수들

폭락, 급락, 급등, 반등, 조정, 과열, 버블, 패닉, 랠리

3) 심리 바꾸는 키워드

금리, 기준금리, 인상, 인하, 물가, 인플레이션, 환율, 달러, 연준, FOMC

4) 위기, 불확실성 키워드

위기, 침체, 불황, 부도, 파산, 금융위기, 쇼크
: 감성 점수의 분산 키워줌 → 공포 강도를 잘 나타내지 않을까??

생각해보면 좋을 점

Q. 우리 모델의 타겟변수는?

Q. 댓글 키워드를 거시적으로 가야 할지, 특정 종목에 관여되는 것으로 가야 할지?