FoxScore Methodology Paper

Methodology, metrics and scoring logic (0–100)
Example calculations fully traceable using Apple Inc. (AAPL)

This paper explains how FoxScore derives financial metrics from price data and then converts them into a standardized 0–100 point scale. The goal is comparability across different assets—with calculation steps that can be followed step by step even without prior knowledge.

What you'll find here

- the data basis and definitions (reference date, time windows, returns)
- the metrics, including formulas and example calculations
- the translation of raw values into scores (0–100) as well as verbal labels
- the aggregation into performance, stability and trend, and the total score
- a fully worked example for Apple (AAPL)

Note / Disclaimer
FoxScore is an analysis and comparison tool and does not constitute investment advice. All calculations are based on historical data. Past performance is not a reliable indicator of future results.

Contents:

1 WHAT IS FOXSCORE? 3
1.1 DATA BASIS 3
1.2 METRICS 3
1.3 THE THREE SUB-SCORES 3
1.4 THE TOTAL SCORE 3
1.5 VERBAL LABELS 4
1.6 IMPORTANT NOTES ON INTERPRETATION 4
2 CALCULATION EXAMPLE (APPLE) 5
2.1 THE PERFORMANCE SCORE 5
2.1.1 Return (total return) 6
2.2 THE STABILITY SCORE 10
2.2.1 Current drawdown 10
2.2.2 Max drawdown (1Y–10Y) 11
2.2.3 Volatility 1Y (annualized) 13
2.2.4 Sharpe Ratio (90d) 14
2.2.5 Sortino Ratio (90d) 15
2.2.6 Return/Vol Ratio (1Y) 16
2.2.7 CAGR/Drawdown Ratio (10Y) 17
2.3 THE TREND SCORE 19
2.3.1 Price 19
2.3.2 SMA 50 19
2.3.3 SMA 100 20
2.3.4 SMA 200 21
2.3.5 Price vs SMA 50 21
2.3.6 Price vs SMA 100 22
2.3.7 Price vs SMA 200 23
2.3.8 Trend Strength 24
2.3.9 Momentum 12M 25
2.3.10 Relative Strength 12M 26
2.3.11 Golden Cross 27
2.3.12 Death Cross 27

1 What is FoxScore?

FoxScore is a comparison tool for assets (e.g., stocks, ETFs, commodities, crypto). Instead of interpreting many different metrics individually, the values are converted into standardized scores (0 to 100). 0 is weak, 50 is average, and 100 is top. The scores are relative to the FoxScore universe: a higher score means a better ranking compared with other assets. The rating scale therefore moves dynamically with the market and shows where assets currently stand technically relative to one another.

1.1 Data basis

The data basis consists of price data, updated three times per day after the close of the major exchanges in the US, Europe and Asia. For stocks, the “Adjusted Close” is typically used (price adjusted for splits and—if included by the data provider—dividend effects). All metrics in this paper can be derived from this price column.

1.2 Metrics

From the market data, 26 established metrics from quantitative financial analysis are computed. To make them comparable, points are assigned: the worst value defines 0 points and the best value 100 points. To make individual metrics easier to interpret, they are labeled as very weak/weak/neutral/strong/very strong. The calculation and scoring are explained in more detail in the following chapters.

1.3 The three sub-scores

The metrics are grouped into three main areas: performance, stability and trend. After applying defined weights to the metrics, each area again yields a comparable scale between 0 and 100.

1.4 The total score

To compare assets at a glance, they also receive an overall rating. The logic is the same as for the metrics: the main groups also receive defined weights, and the total score is calculated from the respective point values.

1.5 Verbal labels

The points are translated into five classes:

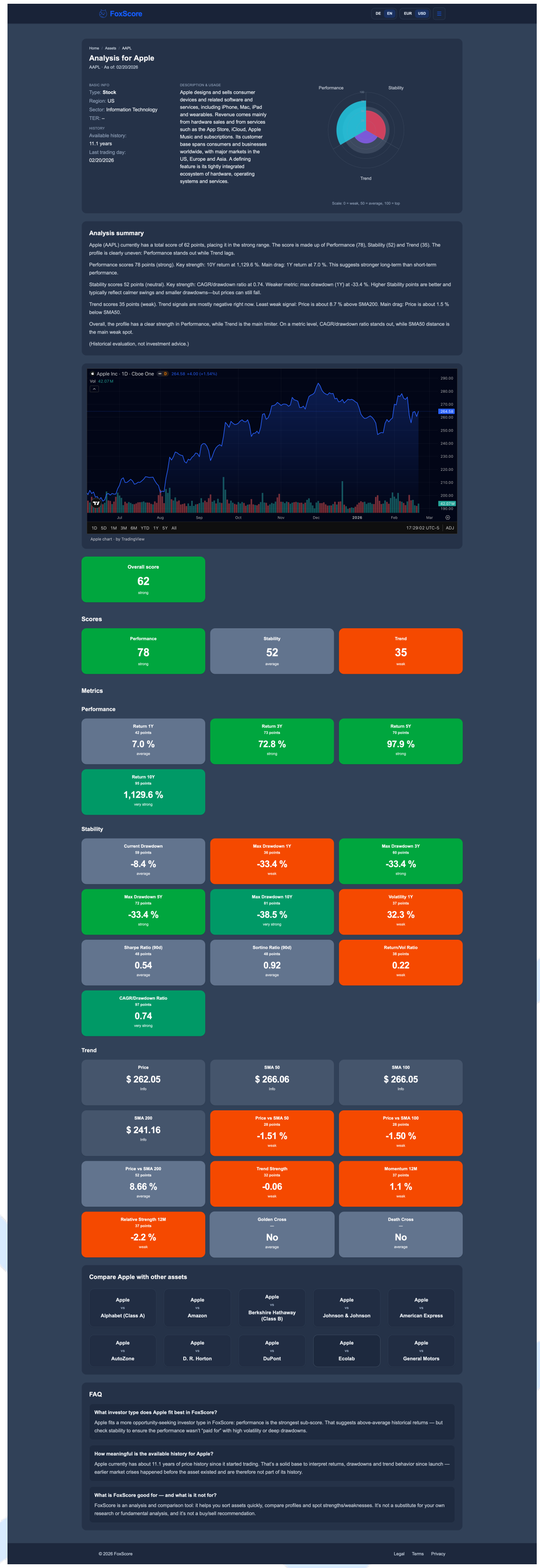
- 0–19 = very weak
- 20–39 = weak
- 40–59 = neutral
- 60–79 = strong
- 80–100 = very strong

1.6 Important notes on interpretation

FoxScore evaluates assets based on historical price data. The scores are relative rankings within the FoxScore universe and can change even if the price of a single asset barely moves, because the comparison set changes. For new assets or instruments with a short history, not all time windows are always available; in aggregations, only available values are therefore taken into account. FoxScore is not a forecast and not investment advice.

2 Calculation example (Apple)

In the screenshot you can see the asset page for Apple (AAPL) with data as of 02/20/2026 in USD. The following sections explain every metric shown and reproduce the calculation using Apple's values.



2.1 The Performance Score

In FoxScore, **performance** describes an asset's historical price development—i.e., how the price has changed over a certain period. Because a single period can easily provide a distorted picture, we look at multiple horizons: short horizons reflect current market phases, while longer horizons smooth short-term outliers and reveal whether a development was sustainable across different market phases. The basis is transparent **total returns**: we compare the price on the reference date with the price at the start of the respective time window and calculate the percentage change. We then place these raw values on a uniform scale so results are easier to compare across different assets. Important: performance evaluates the past and is no guarantee of future returns.

2.1.1 Return

The return over a period is the percentage change between the start and end price.

```
Return = (P_end / P_start) - 1

Return% = Return * 100
```

metric	Endpreis (P_end)	Startpreis (P_start)	Rechenschritt	raw value
Return 1Y	262.05	20.02.2025 / 244.8700	(262.05 / 244.8700 − 1) · 100	7.0%
Return 3Y	262.05	20.02.2023 / 151.6710	(262.05 / 151.6710 − 1) · 100	72.8%
Return 5Y	262.05	20.02.2021 / 132.4210	(262.05 / 132.4210 − 1) · 100	97.9%
Return 10Y	262.05	20.02.2016 / 21.3115	(262.05 / 21.3115 − 1) · 100	1,129.6%

Calculating the performance score

The score is intended to put more weight on long-term returns than on short-term swings. Therefore, a weighted mean is calculated from the four return scores—with higher weight for longer time windows:

1Y = 10%

3Y = 20%

5Y = 30%

10Y = 40%

To make the performance score comparable with the other assets in the FoxScore universe, the raw values must be translated. We do not look only at the raw value itself, but at how it stands relative to all other assets in the same period:

$$S3(t) = s\_high(\text{ret\_3y}(t))$$
$$S5(t) = s\_high(\text{ret\_5y}(t))$$
$$S10(t) = s\_high(\text{ret\_10y}(t))$$

Step 1: Define the "universe"

For each time window (e.g., 1 year), we consider all assets that have a valid value for that window.

Example for 1Y on the reference date: t

$$U_{1Y}(t) = \{\text{ret}_{1Y}(t) \text{ aller Assets mit gültigem 1Y-Return}\}$$

Where:
U = set of all comparison values (the universe)
$N = | U |$ = number of assets in this universe

Step 2: Determine rank position

We sort all values in the universe *U* from worst to best.
Then we determine the rank position of the value *x* under consideration:

r(x) = 0 means: the worst value in the universe
r(x) = N − 1 means: the best value in the universe

Step 3: Rank → Score (0–100)

FoxScore converts raw values into points via a percentile rank (0–100). To do this, on the reference date all valid values of a metric are sorted in ascending order.
For a value x, idx is the position of the last value ≤ x ("rightmost ≤ x"). With N = the number of valid values, the percentile is p = idx/(N−1) (p ∈ [0,1]).
Then: if higher is better → Score = round(100 · p); if lower is better → Score = round(100 · (1−p)). If N < 2, no score is calculated (Score = NULL); the metric is dropped on the reference date and is not considered in aggregations.
Note: If you want to reproduce p exactly, you need N and idx from the FoxScore universe on the reference date (see appendix: SQL query).

Sub-score	Input (raw value)	Step 1: Comparison set (FoxScore universe)	Step 2: Determine percentile	Step 3: Perzentil → Score	Result
S1	Return 1Y = 7.0%	All assets with a valid **1Y return**	p (see Appendix: idx/N)	S1 ≈ round(100 · p)	42
S3	Return 3Y = 72.8%	All assets with a valid **3Y return**	p (see Appendix: idx/N)	S3 ≈ round(100 · p)	73
S5	Return 5Y = 97.9%	All assets with a valid **5Y return**	p (see Appendix: idx/N)	S5 ≈ round(100 · p)	70
S10	Return 10Y = 1,129.6%	All assets with a valid **10Y return**	p (see Appendix: idx/N)	S10 ≈ round(100 · p)	95

From the returns, Apple has the following sub-scores (0–100):

$$S1 = 42(\text{Return 1Y})$$
$$S3 = 73(\text{Return 3Y})$$
$$S5 = 70(\text{Return 5Y})$$
$$S10 = 95(\text{Return 10Y})$$

Final formula for the performance score:

$$PerformanceScore = 0.10 \cdot S1 + 0.20 \cdot S3 + 0.30 \cdot S5 + 0.40 \cdot S10$$

Now we plug the values into the weighting:
Note: The weighting is computed using decimals; only the final result is rounded to whole points and then clipped to 0–100 (e.g., 77.8 → 78).

Zeitraum	Teilscore	Gewicht	Beitrag
1Y	42	0.10	4.2
3Y	73	0.20	14.6
5Y	70	0.30	21.0
10Y	95	0.40	38.0
	Summe		**77.8**

$$PerformanceScore_{Apple} = 77.8 \approx 78$$

Conclusion: In the performance comparison, Apple achieves a score of 78 and thus falls into the “strong” category. (See p.4: “1.5 Verbal labels”.)

Not every asset has sufficient history for all horizons (e.g., 10 years). In that case, missing sub-values are not artificially filled. Instead, we compute the score as a weighted mean over the available sub-values. The weights used are normalized to the available time windows (division by the sum of the weights used).

2.2 The Stability Score

The stability score answers the following question:
How calm and resilient is an asset—measured by volatility and drawdowns?
While the performance score focuses on return, stability evaluates the side effects of those returns: How strongly does the price fluctuate? How deep and how frequent are drawdowns? A high stability score therefore means: lower volatility, smaller/rarer drawdowns, and a better return-to-risk relationship.

Important:
Stability is **not a forecast** of the future.
It describes the **past** over defined time windows (90/252/365 days or 3/5/10 years).

Some metrics are “lower is better” (e.g., volatility, drawdown), others “higher is better” (e.g., Sharpe, Return/Vol). Because higher values should correspond to a better score, “lower is better” metrics are inverted.

2.2.1 Current drawdown

The current drawdown measures how far the price on the reference date is below its previous all-time high (ATH). A value of 0% means: price at ATH. Negative values mean: price below ATH. The closer the value is to 0, the better.

$$dd\_current = (P\_end / P\_ATH) - 1$$
$$dd\_current\% = dd\_current * 100$$

Step 1: Determine the end price on the reference date

Price value on the reference date as P_end.

Step 2: Determine the all-time high up to the reference date

Determine the previous all-time high (including the reference date) as P_ATH = max(P_t) for all days t ≤ reference date. The ATH date t_ATH is the last day on which this P_ATH was reached.

Step 3: Calculate the current drawdown

Insert P_end and P_ATH into the formula and convert the result to percent (· 100).

metric	End price (P_end)	ATH price	Calculation	raw value
Current drawdown	262.05	286.19	(262.05 / 286.19 − 1) · 100	−8.4%

Score conversion (0–100 points)

To make the raw value comparable, dd_current on the reference date is ranked within the FoxScore universe of all assets with a valid dd_current (here: N = 887). Because a value closer to 0 is better, higher dd_current values are better.
FoxScore derives a rank position (bad → good) and converts it into a score from 0 to 100 points (more points = better). In the UI, Apple reaches 55 points for the current drawdown.

Conclusion: On the reference date, Apple is −8.4% below its previous ATH and is rated "neutral" at 55 points.

2.2.2 Max drawdown (1Y–10Y)

Max drawdown measures the largest price decline within a fixed time window. It describes how far the price fell from the highest level (ATH within the window) to a later low. The value is typically negative. The closer it is to 0% (i.e., the less negative), the better.

FoxScore computes max drawdowns in fixed windows of 1Y, 3Y, 5Y and 10Y. A window is considered available only if at least 90% of the required data points are present.

$$roll\_max_t = \max(P_u)$$
$$dd_t = (P_t / roll\_max_t) - 1$$
$$maxdd_W = \min(dd_t)$$
$$maxdd\% = maxdd_W \cdot 100$$

Step 1: Set the window length

For stocks, per_year = 252 trading days (for crypto per_year = 365 calendar days). This yields W = per_year · {1, 3, 5, 10}. For the calculation, at least ceil(0.90 · W) observations are required.

Step 2: Determine rolling ATH within the window

For each day t, compute the rolling high over the last W days: roll_max_t.

Step 3: Calculate the drawdown series

For each day t, compute the drawdown relative to this window ATH:
dd_t = (P_t / roll_max_t) − 1.

Step 4: Determine the worst drawdown in the window

The max drawdown is the minimum of the dd_t values in the window considered.

Step 5: In Prozent umrechnen

Multiply the raw value by 100 to obtain percentage values.

metric	**Peak (date/price)**	**Low (date/price)**	**Rechenschritt**	**raw value**
Max Drawdown 1Y	26.12.2024 / 258.736	08.04.2025 / 172.42	(172.42 / 258.736 − 1) · 100	−33.4%
Max Drawdown 3Y	26.12.2024 / 258.736	08.04.2025 / 172.42	(172.42 / 258.736 − 1) · 100	−33.4%
Max Drawdown 5Y	26.12.2024 / 258.736	08.04.2025 / 172.42	(172.42 / 258.736 − 1) · 100	−33.4%
Max Drawdown 10Y	03.10.2018 / 55.1624	03.01.2019 / 33.9151	(33.9151 / 55.1624 − 1) · 100	−38.5%

Note on interpreting the peak/low table: The worst drawdown is determined over the last W days. For the day of the low point, the associated rolling ATH over the preceding W days is considered. As a result, the peak date (ATH) can lie before the start of the reference-date window, even though the low point lies within the window.

Score conversion (0–100 points)

To make the raw value comparable, maxdd_W on the reference date is ranked within the FoxScore universe of all assets with a valid maxdd_W. Because smaller drawdowns are better, higher maxdd_W values (closer to 0) are better.
FoxScore derives a rank position (bad → good), computes a percentile p and converts it into a score: Score ≈ round(100 · p).

metric	Input (raw value)	Step 1: Comparison set (FoxScore universe)	Step 2: Determine percentile	Step 3: Perzentil → Score	Ergebnis
Max Drawdown 1Y	−33.4%	N = 792	p (see Appendix: idx/N)	Score ≈ round(100 · p)	36 (weak)
Max Drawdown 3Y	−33.4%	N = 788	p (see Appendix: idx/N)	Score ≈ round(100 · p)	60 (strong)
Max Drawdown 5Y	−33.4%	N = 776	p (see Appendix: idx/N)	Score ≈ round(100 · p)	72 (strong)
Max Drawdown 10Y	−38.5%	N = 700	p (see Appendix: idx/N)	Score ≈ round(100 · p)	81 (very strong)

Conclusion: In the 1Y window, Apple shows a pronounced historical setback (−33.4%) and is rated "weak" at 36 points. Over longer horizons (3Y/5Y), the maximum drop is relatively better (60–74 points, "strong"). In the 10Y window, the maximum decline is −38.5% and reaches 81 points ("very strong").

2.2.3 Volatility 1Y (annualized)

Volatility measures the typical amplitude of an asset's fluctuations. In FoxScore, volatility 1Y is the annualized standard deviation of daily returns over the last 1Y window. Lower is better.

```
ret_1d(t) = (P_t / P_{t-1}) - 1
vol_1Y = Std(ret_1d) over the last per_year days
vol_1Y_ann = vol_1Y · √per_year
vol_1Y_ann% = vol_1Y_ann · 100
```

Step 1: Calculate daily returns

For each trading day, the daily return ret_1d from the closing prices.

Step 2: Standard deviation in the 1Y window

Standard deviation of the last per_year daily returns (for stocks per_year = 252).

Step 3: Annualize and convert to percent

Multiply by √per_year and convert to percent (· 100).

metric	Input	Reference values	Calculation	raw value
Volatility 1Y (daily Std)	ret_1d	per_year = 252	Std(ret_1d)	0.020316
Volatility 1Y (annualized)	vol_1Y	√252	vol_1Y · √per_year	0.322505
Volatility 1Y (%)	vol_1Y_ann		vol_1Y_ann · 100	32.3%

Score conversion (0–100 points)

FoxScore ranks the raw value on the reference date within the universe of all assets with a valid value (N = 888). Because lower is better, the percentile rank p is computed according to the section “Rank → Score” and converted into points. In the UI, Apple reaches 37 points.

Conclusion: Apple reaches 37 points for volatility 1Y and is rated “weak”.

2.2.4 Sharpe Ratio (90d)

The Sharpe ratio relates return to overall volatility. In FoxScore, Sharpe (90d) is time-consistent (not additionally annualized): 90-day total return divided by 90-day volatility. Higher is better.

```
roll_ret_90 = (∏_{i=1..90} (1 + ret_1d(i))) - 1

roll_vol_90 = Std(ret_1d) over 90 days · √90

sharpe_90d = roll_ret_90 / roll_vol_90
```

Step 1: 90-day total return

For the 90-day total return, the return over the entire 90-day window is understood as cumulative growth: from the daily returns, a growth series is formed by using (1 + ret_1d) for each day and multiplying these factors across 90 days. Subtracting 1 then yields the window’s total return (roll_ret_90).

Step 2: 90-day volatility

The 90-day volatility describes the typical fluctuation range of daily returns within the same window. The standard deviation of daily returns is calculated over 90 days. To scale the fluctuation to the window length, this daily standard deviation is multiplied by √90; this yields the window volatility (roll_vol_90).

Step 3: Quotient bilden

The Sharpe ratio (90d) relates these two quantities: it is defined as the quotient (sharpe_90d = roll_ret_90 / roll_vol_90) and thus expresses how much 90-day return was achieved per unit of 90-day volatility.

metric	Input	Reference values	Calculation	raw value
90d-Gesamtrendite	ret_1d	window = 90	∏(1+ret_1d) − 1	0.068414
90d volatility	ret_1d	window = 90	Std(ret_1d) · √90	0.127300
Sharpe (90d)	roll_ret_90, roll_vol_90		roll_ret_90 / roll_vol_90	0.537427

Score conversion (0–100 points)

FoxScore ranks the raw value on the reference date within the universe of all assets with a valid value (N = 888). Because higher is better, the percentile rank p is computed according to the section "Rank → Score" and converted into points. In the UI, Apple reaches 49 points.

Conclusion: Apple reaches 49 points for Sharpe (90d) and is rated "neutral".

2.2.5 Sortino Ratio (90d)

The Sortino ratio is a Sharpe variant that counts only negative volatility ("downside") as risk. In FoxScore: 90-day total return divided by 90-day downside volatility. Higher is better.

```
neg_ret = min(ret_1d, 0)

roll_downside_vol_90 = Std(neg_ret) over 90 days · √90

sortino_90d = roll_ret_90 / roll_downside_vol_90
```

Step 1: Negative Tagesrenditen isolieren

For the downside view, the daily return is first transformed so that only negative moves count as risk: positive daily returns are set to 0, while negative daily returns are kept unchanged. This transformed series is called neg_ret and thus captures only the "downward components" of daily fluctuations.

Step 2: Downside volatility (90d)

The 90-day downside volatility then describes the typical magnitude of these negative swings within the 90-day window. The standard deviation of neg_ret is calculated over 90 days. To scale it to the window length, it is multiplied by √90—analogous to overall volatility—yielding roll_downside_vol_90.

Step 3: Calculate the ratio

The Sortino ratio (90d) then relates the 90-day total return (roll_ret_90) to this downside volatility: defined as the quotient (sortino_90d = roll_ret_90 / roll_downside_vol_90), it shows how much return per unit of "negative risk" was achieved in the 90-day window.

metric	Input	Reference values	Calculation	raw value
90d-Gesamtrendite	ret_1d	window = 90	∏(1+ret_1d) − 1	0.068414
90d downside volatility	neg_ret	window = 90	Std(neg_ret) · √90	0.074232
Sortino (90d)	roll_ret_90, roll_downside_vol_90		roll_ret_90 / roll_downside_vol_90	0.921631

Score conversion (0–100 points)

FoxScore ranks the raw value on the reference date within the universe of all assets with a valid value (N = 888). Because higher is better, the percentile rank p is computed according to the section "Rank → Score" and converted into points. In the UI, Apple reaches 48 points.

Conclusion: Apple reaches 48 points for Sortino (90d) and is rated "neutral".

2.2.6 Return/Vol Ratio (1Y)

The Return/Vol Ratio relates the 1Y total return to the annualized 1Y volatility. It measures how much return was achieved per unit of volatility. Higher is better.

```
return_vol_ratio = ret_1y / vol_1Y_ann
```

Step 1: Determine 1Y return

For the Return/Vol Ratio, the 1Y total return is used as the starting point. It describes how much the price changed over the 1-year window and is available as ret_1y.

Step 2: Determine 1Y volatility (annualized)

As the second component, the annualized 1Y volatility (vol_1Y_ann) is used. It summarizes the typical fluctuation range of daily returns over the same one-year window and thus measures how "expensive" that return was in terms of volatility.

Step 3: Calculate the ratio

The metric itself is the ratio of both quantities: return_vol_ratio = ret_1y / vol_1Y_ann. It expresses how much one-year return was achieved per unit of annualized volatility—the higher this value, the more efficient the return relative to risk.

metric	Input	Reference values	Calculation	raw value
Return 1Y	P_end, P_start	per_year = 252	(P_end / P_start) − 1	0.070160
Volatility 1Y (ann.)	ret_1d		Std(ret_1d) · √per_year	0.322505
Return/Vol Ratio	ret_1y, vol_1Y_ann		ret_1y / vol_1Y_ann	0.217546

Score conversion (0–100 points)

FoxScore ranks the raw value on the reference date within the universe of all assets with a valid value (N = 888). Because higher is better, the percentile rank p is computed according to the section “Rank → Score” and converted into points. In the UI, Apple reaches 38 points.

Conclusion: Apple reaches 38 points for Return/Vol Ratio and is rated “weak”.

2.2.7 CAGR/Drawdown Ratio (10Y)

The CAGR/Drawdown ratio combines long-term return with maximum drawdown: it relates the annualized 10Y return (CAGR) to the maximum drawdown over 10 years. Higher is better.

```
CAGR_10Y = (1 + ret_10y)^(1/10) - 1

cagr_drawdown_ratio = CAGR_10Y / |maxdd_10y|
```

Step 1: Determine 10Y total return

As a starting point, the 10Y total return (ret_10y) is used. It describes how much the asset developed overall over the entire 10-year window.

Step 2: Calculate CAGR

To make this long-term development comparable with other periods, it is converted into an annualized return: the CAGR (10Y). It corresponds to the constant annual return which—compounded over 10 years—leads exactly to the observed 10Y total return. Formally: CAGR_10Y = (1 + ret_10y)^(1/10) − 1.

Step 3: Durch maximalen Drawdown teilen

Next, this annualized return is related to the risk captured over the same period by the maximum drawdown (10Y). Because drawdown is a negative value, its absolute value is used. The metric is therefore cagr_drawdown_ratio = CAGR_10Y / |maxdd_10y| and expresses how much annualized return was achieved per unit of maximum decline.

metric	Input	Reference values	Calculation	raw value
Return 10Y	P_end, P_start	window = 10Y	(P_end / P_start) − 1	11.296178
CAGR 10Y	ret_10y	years = 10	(1+ret_10y)^(1/10) − 1	0.285219
Max Drawdown 10Y	price-Serie	Window = 10Y	min(dd_t)	-0.385177
CAGR/Drawdown Ratio	CAGR_10Y, maxdd_10y		CAGR_10Y / |maxdd_10y|	0.740487

Score conversion (0–100 points)

FoxScore ranks the raw value on the reference date within the universe of all assets with a valid value (N = 888). Because higher is better, the percentile rank p is computed according to the section “Rank → Score” and converted into points. In the UI, Apple reaches 97 points.

Conclusion: Apple reaches 97 points for CAGR/Drawdown Ratio and is rated “very strong”.

2.3 The Trend Score

2.3.1 Price

The price is the price value on the reference date. In the formulas it is denoted as the end price P_end.
The price is a base input (e.g., for returns and trend metrics). On its own, neither "higher" nor "lower" is inherently better.

```
P_end = P_t  (price on the reference date)
```

Step 1: Define the reference date

The calculation always refers to a specifically defined reference date (in the example: 20.02.2026). This reference date determines which data point from the time series is used as the reference.

Step 2: Determine the end price on the reference date

From the price time series, the price value exactly on the reference date is taken and defined as P_end (or P_t). P_end is therefore the baseline value to which further comparisons (e.g., against moving averages) refer.

Score conversion (0–100 points)

In FoxScore, the price is not converted into points (0–100) as its own metric.
It is shown in the UI as the base value and serves as an input for further metrics (e.g., SMA, Price vs SMA).

Conclusion: On the reference date 20.02.2026, Apple (AAPL) has a price of 262.05.

2.3.2 SMA 50

The Simple Moving Average (SMA) with a window length of 50 is the simple average of the last 50 price values up to and including the reference date.
The SMA smooths short-term fluctuations and serves as a reference line for trend comparisons.

```
SMA_50 = (1/50) · Σ_{i=0..49} P_{t-i}
```

Step 1: Set the window length

For calculating SMA 50, the window length is fixed and equals 50 days.

Step 2: Determine the price window

The data basis is a rolling 50-day window from the price time series: the values P_{t-49}, …, P_t, where P_t is the price on the reference date.

Step 3: Durchschnitt bilden

From these 50 prices, the arithmetic mean is computed. This mean is SMA_50.

Note on data coverage

In the code, SMA_50 is calculated as rolling(50).mean(). This means: at least 50 price values are required; otherwise SMA_50 is not defined (NaN).

metric	**Input**	**Reference values**	**Calculation**	**raw value**
SMA 50	price	window = 50	(1/50) · Σ P_(t-i)	266.0625

Score conversion (0–100 points)

SMA 50 is not converted into points (0–100) as a standalone metric in FoxScore.
It is shown as the base value and feeds into derived trend metrics (e.g., Price vs SMA).

Conclusion: On the reference date 20.02.2026, Apple (AAPL) has an SMA 50 of 266.062500.

2.3.3 SMA 100

The Simple Moving Average (SMA) with a window length of 100 is the simple average of the last 100 price values up to and including the reference date.
The SMA smooths short-term fluctuations and serves as a reference line for trend comparisons.

$$SMA\_100 = (1/100) \cdot \Sigma\_\{i=0..99\}\ P\_\{t-i\}$$

Step 1: Set the window length

For calculating SMA 100, the window length is fixed and equals 100 days.

Step 2: Determine the price window

The data basis is a rolling 100-day window from the price time series: the values P_{t-99}, …, P_t, where P_t is the price on the reference date.

Step 3: Compute the average

From these 100 prices, the arithmetic mean is computed. This mean is SMA_100.

Note on data coverage

In the code, SMA_100 is calculated as rolling(100).mean(). This means: at least 100 price values are required; otherwise SMA_100 is not defined (NaN).

metric	**Input**	**Reference values**	**Calculation**	**raw value**
SMA 100	price	window = 100	(1/100) · Σ P_(t-i)	266.05255

Score conversion (0–100 points)

SMA 100 is not converted into points (0–100) as a standalone metric in FoxScore.
It is shown as the base value and feeds into derived trend metrics (e.g., Price vs SMA).

Conclusion: On the reference date 20.02.2026, Apple (AAPL) has an SMA 100 of 266.052550.

2.3.4 SMA 200

The Simple Moving Average (SMA) with a window length of 200 is the simple average of the last 200 price values up to and including the reference date.
The SMA smooths short-term fluctuations and serves as a reference line for trend comparisons.

$$SMA\_200 = (1/200) \cdot \Sigma_{i=0..199} P_{t-i}$$

Step 1: Set the window length

For calculating SMA 200, the window length is fixed and equals 200 days.

Step 2: Determine the price window

The data basis is a rolling 200-day window from the price time series: the values P_{t-199}, …, P_t, where P_t is the price on the reference date.

Step 3: Compute the average

From these 200 prices, the arithmetic mean is computed. This mean is SMA_200.

Note on data coverage

In the code, SMA_200 is calculated as rolling(200).mean(). This means: at least 200 price values are required; otherwise SMA_200 is not defined (NaN).

metric	**Input**	**Reference values**	**Calculation**	**raw value**
SMA 200	price	window = 200	(1/200) · Σ P_(t-i)	241.163225

Score conversion (0–100 points)

SMA 200 is not converted into points (0–100) as a standalone metric in FoxScore.
It is shown as the base value and feeds into derived trend metrics (e.g., Price vs SMA).

Conclusion: On the reference date 20.02.2026, Apple (AAPL) has an SMA 200 of 241.163225.

2.3.5 Price vs SMA 50

This metric measures the relative distance of the price from the 50-day moving average (SMA 50). A higher value means the price is further above its short-term average; a negative value means the price is below it.

$$SMA\_50(t) = (1/50) \cdot \Sigma_{i=0..49} P_{t-i}$$

$$price\_vs\_sma50(t) = (P_t / SMA\_50(t)) - 1$$

$$price\_vs\_sma50\%(t) = price\_vs\_sma50(t) \cdot 100$$

Step 1: Determine reference date and end price

The calculation refers to a reference date t. From the price time series, the price on the reference date is taken as P_t.

Step 2: Calculate SMA 50

For SMA 50, a rolling window of 50 prices is used: P_{t-49}, …, P_t. SMA 50 is the arithmetic mean of these 50 values.

Step 3: Calculate relative distance

The distance is computed as the ratio of price to SMA: (P_t / SMA_P_t / SMA_50) − 1. For the UI, the raw value is also shown in percent (· 100).

metric	**Input**	**Reference values**	**Calculation**	**raw value**
SMA 50	Prices P_{t-49} … P_t	Window length = 50	arithm. mean	266.0625
Price vs SMA 50	P_t = 262.05, SMA_50	—	(P_t / SMA_50) − 1	−0.0150810429880197
Price vs SMA 50 (%)	price_vs_sma50	· 100	price_vs_sma50 · 100	−1.50810429880197%

Score conversion (0–100 points)

FoxScore ranks the raw value on the reference date within the universe of all assets with a valid value (N = 792). Because higher is better, the percentile rank p is computed according to the section "Rank → Score" and converted into points. In the UI, Apple reaches 28 points.

Conclusion: On the reference date, Apple is -1.51% below its SMA 50 and reaches 28 points ("weak").

2.3.6 Price vs SMA 100

This metric measures the relative distance of the price from the 100-day moving average (SMA 100). A higher value means the price is further above its medium-term average; a negative value means the price is below it.

```
SMA_100(t) = (1/100) · Σ_{i=0..99} P_{t-i}

price_vs_sma100(t) = (P_t / SMA_100(t)) - 1

price_vs_sma100%(t) = price_vs_sma100(t) · 100
```

Step 1: Determine reference date and end price

The calculation refers to a reference date t. From the price time series, the price on the reference date is taken as P_t.

Step 2: Calculate SMA 100

For SMA 100, a rolling window of 100 prices is used: P_{t-99}, …, P_t. SMA 100 is the arithmetic mean of these 100 values.

Step 3: Calculate relative distance

The distance is computed as the ratio of price to SMA: (P_t / SMA_P_t / SMA_100) − 1. For the UI, the raw value is also shown in percent (· 100).

metric	Input	Reference values	Calculation	raw value
SMA 100	Prices P_{t-99} … P_t	Window length = 100	arithm. mean	266.05255
Price vs SMA 100	P_t = 262.05, SMA_100	—	(P_t / SMA_100) − 1	−0.0150442083716168
Price vs SMA 100 (%)	price_vs_sma100	· 100	price_vs_sma100 · 100	−1.50442083716168%

Score conversion (0–100 points)

FoxScore ranks the raw value on the reference date within the universe of all assets with a valid value (N = 792). Because higher is better, the percentile rank p is computed according to the section "Rank → Score" and converted into points. In the UI, Apple reaches 28 points.

Conclusion: On the reference date, Apple is -1.50% below its SMA 100 and reaches 28 points ("weak").

2.3.7 Price vs SMA 200

This metric measures the relative distance of the price from the 200-day moving average (SMA 200). A higher value means the price is further above its long-term average; a negative value means the price is below it.

```
SMA_200(t) = (1/200) · Σ_{i=0..199} P_{t-i}

price_vs_sma200(t) = (P_t / SMA_200(t)) - 1

price_vs_sma200%(t) = price_vs_sma200(t) · 100
```

Step 1: Determine reference date and end price

The calculation refers to a reference date t. From the price time series, the price on the reference date is taken as P_t.

Step 2: Calculate SMA 200

For SMA 200, a rolling window of 200 prices is used: P_{t-199}, …, P_t. SMA 200 is the arithmetic mean of these 200 values.

Step 3: Calculate relative distance

The distance is computed as the ratio of price to SMA: (P_t / SMA_P_t / SMA_200) − 1. For the UI, the raw value is also shown in percent (· 100).

metric	Input	Reference values	Calculation	raw value
SMA 200	Prices P_{t-199} … P_t	Window length = 200	arithm. mean	241.163225
Price vs SMA 200	P_t = 262.05, SMA_200	—	(P_t / SMA_200) − 1	0.0866084578193878
Price vs SMA 200 (%)	price_vs_sma200	· 100	price_vs_sma200 · 100	8.66084578193878%

Score conversion (0–100 points)

FoxScore ranks the raw value on the reference date within the universe of all assets with a valid value (N = 792). Because higher is better, the percentile rank p is computed according to the section "Rank → Score" and converted into points. In the UI, Apple reaches 52 points.

Conclusion: On the reference date, Apple is 8.66% above its SMA 200 and reaches 52 points ("neutral").

2.3.8 Trend Strength

Trend Strength measures how "straight" a trend runs over a fixed window. FoxScore computes the correlation between log(price) and time in the rolling window. The value lies in [-1, +1]: near +1 means a stable uptrend, near −1 a stable downtrend, near 0 means little linear trend.

```
logP(t) = ln(P_t)  (only for P_t > 0)

trend_strength(t) = corr( logP in window t-89..t , time index 0..89 )
```

Step 1: Define the price window

A rolling window of 90 days up to and including the reference date is used: P_{t-89}, …, P_t.

Step 2: Build log prices and time index

For all prices in the window, the natural logarithm is computed (ln). In parallel, a time index 0, 1, … is created.

Step 3: Calculate correlation

Trend Strength is the Pearson correlation between log prices and the time index within the window. The result is clipped to [-1, +1].

metric	Input	Reference values	Calculation	raw value
Trend Strength	Prices P_{t-89} … P_t	Window length = 90	corr(ln(P), Time index)	−0.0585546166215596

Score conversion (0–100 points)

FoxScore ranks the raw value on the reference date within the universe of all assets with a valid value (N = 792). Because higher is better, the percentile rank p is computed according to the section "Rank → Score" and converted into points. In the UI, Apple reaches 32 points.
Conclusion: On the reference date, Apple has a Trend Strength of -0.06 and reaches 32 points ("weak").

2.3.9 Momentum 12M

Momentum 12M in FoxScore is a 12-1 momentum: the return from "12 months ago" to "1 month ago". The last month is omitted to weight short-term reversals less. A higher value is better.

```
mom12_skip = round(mom12_days / 12)

momentum_12m(t) = (P_{t-mom12_skip} / P_{t-mom12_days}) - 1

momentum_12m%(t) = momentum_12m(t) · 100
```

Step 1: Set the window length

For traditional markets, mom12_days = 252 (trading days) is used. The skip is mom12_skip = round(252/12) = 21 trading days.

Step 2: Determine comparison prices

Required are P_{t-21} (here: 2026-01-21 / 247.65) and P_{t-252} (here: 2025-02-19 / 244.87).

Step 3: Calculate return and convert to percent

Compute (P_{t-skip} / P_{t-12m}) − 1 and convert to percent for the UI (· 100).

metric	Input	Reference values	Calculation	raw value
Momentum 12M	P_{t-21}, P_{t-252}	skip = 21, 12M = 252 days	(P_{t-skip} / P_{t-12m}) − 1	0.0113529627965860
Momentum 12M (%)	momentum_12m	· 100	momentum_12m · 100	1.1352962796586%

Score conversion (0–100 points)

FoxScore ranks the raw value on the reference date within the universe of all assets with a valid value (N = 792). Because higher is better, the percentile rank p is computed according to the section "Rank → Score" and converted into points. In the UI, Apple reaches 37 points.

Conclusion: Apple has a 12-1 momentum of 1.14% and reaches 37 points ("weak").

2.3.10 Relative Strength 12M

Relative Strength 12M compares an asset's 12-1 momentum with the momentum of a benchmark. If the value is positive, the asset outperformed the benchmark in momentum; if it is negative, it was weaker. If benchmark_symbol is not set (NULL), FoxScore uses a default benchmark/fallback from the codebase so that relative strength can still be computed.

```
RS_12M(t) = ((1 + mom_12m_asset(t)) / (1 + mom_12m_bench(t))) - 1

RS_12M%(t) = RS_12M(t) · 100
```

Step 1: Determine asset momentum

First, momentum_12m is calculated for the asset (see section "Momentum 12M").

Step 2: Determine and align benchmark momentum

The benchmark momentum is maintained as a time series and aligned to the asset days via reindex + forward fill. On the asset reference date, the most recently available benchmark value is used.

Step 3: Calculate relative strength

Insert both momentum values into RS_12M = ((1 + mom_a)/(1 + mom_b)) − 1 and convert to percent for the UI (· 100).

metric	Input	Reference values	Calculation	raw value
Relative Strength 12M	mom_a, mom_b (Default-Benchmark/Fallback)	Benchmark-day = 2026-02-19	((1+mom_a)/(1+mom_b)) − 1	−0.0219924242129358
Relative Strength 12M (%)	relative_strength_12m	· 100	relative_strength_12m · 100	−2.19924242129358%

Score conversion (0–100 points)

FoxScore ranks the raw value on the reference date within the universe of all assets with a valid value (N = 792). Because higher is better, the percentile rank p is computed according to the section "Rank → Score" and converted into points. In the UI, Apple reaches 37 points.
Conclusion: Relative to the benchmark (default benchmark), Apple is at −2.20% and reaches 37 points ("weak").

2.3.11 Golden Cross

A Golden Cross in FoxScore is a crossover event: the short-term moving average (SMA 50) moves from below the long-term moving average (SMA 200) to above it. To ensure it is a true crossover (and not just a state that has existed for some time), the previous day is also checked.

```
is_golden_cross(t) = 1, falls SMA_50(t) > SMA_200(t)
AND SMA_50(t-1) ≤ SMA_200(t-1); otherwise 0
```

SMA_50(t) and SMA_200(t) are the arithmetic means of the last 50 and 200 prices up to and including reference date t. The signal is binary (0/1) and is set to 1 only on the day of the crossover.

metric	**Input**	**Reference values**	**Calculation**	**raw value**
Golden Cross	SMA_50(t), SMA_200(t), SMA_50(t-1), SMA_200(t-1)	t-1 = prior day	Check condition → 0/1	0/1

Score conversion (0–100 points)

Golden Cross and Death Cross are not converted into a percentile-based score as standalone metrics. Instead, they act as a bonus or malus on the trend base score on the respective event day.

```
cross_bonus(t) = 6 · is_golden_cross(t) − 6 · is_death_cross(t)
trend_score(t) = clip(round(trend_base(t) + cross_bonus(t)), 0, 100)
```

Conclusion: For Apple, no Golden Cross is active on the reference date (is_golden_cross = 0).

2.3.12 Death Cross

A Death Cross is the counterpart: the short-term moving average (SMA 50) moves from above the long-term moving average (SMA 200) to below it. Here too, the previous day is used to identify the crossover unambiguously.

```
is_death_cross(t) = 1 if SMA_50(t) < SMA_200(t) AND SMA_50(t-1) ≥ SMA_200(t-1);
otherwise 0
```

As with the Golden Cross, the signal is binary (0/1) and is set to 1 only on the crossover day.

metric	Input	Reference values	Calculation	raw value
Death Cross	SMA_50(t), SMA_200(t), SMA_50(t-1), SMA_200(t-1)	t-1 = prior day	Check condition → 0/1	0/1

Score conversion (0–100 points)

The bonus/malus is identical to the “Golden Cross” section and is applied to the trend base score via cross_bonus(t).

```
cross_bonus(t) = 6 · is_golden_cross(t) − 6 · is_death_cross(t)

trend_score(t) = clip(round(trend_base(t) + cross_bonus(t)), 0, 100)
```

Conclusion: For Apple, no Death Cross is active on the reference date (is_death_cross = 0).

3 Appendix: Percentile rank p

The **percentile rank** p describes an asset's **relative position** within the entire comparison universe on the **same reference date**. For a metric, the distribution of all valid values is considered (number of valid values: N) and the asset value x is placed within it.

Formally, p is based on a rank index idx: the position of the last value in the ascendingly sorted distribution that is less than or equal to x ("rightmost ≤ x"). This handles ties consistently: all assets with the same value share the same rank at the upper end of the tie.
This rank yields the percentile:

$$p = \frac{idx}{N - 1}$$

This keeps p in the range [0, 1]: values near 0 correspond to the lower end of the distribution, values near 1 to the upper end.

To convert to the **score (0–100 points)**, p is scaled and rounded. The direction depends on whether "higher" or "lower" is better:

higher is better: Score = $Score = round(100 \cdot p)$
lower is better: Score = $Score = round(100 \cdot (1 - p))$

This converts a metric—independent of unit and magnitude—into a standardized point scale that expresses its **position relative to all other assets.**

Special case: If N < 2 (too few comparison values), the metric is not meaningfully comparable on the reference date. In this case, no score is calculated (Score = NULL) and the metric does not flow into aggregated scores. If an asset's raw value is missing (NaN), the score is also NULL.

Beispiel (Apple, Sharpe Ratio 90d):
On the reference date, Apple has a Sharpe value of 0.54 and reaches 48 points in the UI.
To do this, Apple is placed within the universe of all valid Sharpe values:

```
idx = Anzahl(Sharpe <= 0.54) - 1

p = idx / (N - 1)

Score = round(100 · p)
```

Since Score = 48, p is close to 0.48 (more precisely: p in the interval [0.475, 0.485)). Interpretation: about 48% of valid Sharpe values (including ties) are at or below Apple's value; Apple is therefore roughly in the middle.