Moteur de Scoring Macroéconomique

Architecture, Méthodologie & Validation du Signal

Ce document décrit la logique du notebook `test_engine.ipynb` *—*
pipeline de calcul du score macro et évaluation de sa valeur prédictive.

Contents

Vue d'ensemble

Le notebook implémente un **moteur de scoring macroéconomique** (MODEL_1) qui produit, pour chaque grande zone monétaire, un score composite synthétisant l'état des publications de données économiques. Ce score est ensuite confronté aux rendements de marché pour estimer sa valeur prédictive.

Le pipeline se décompose en quatre étapes principales :

1. **Collecte et traitement des données** — récupération des publications économiques par indicateur et par zone.
2. **Normalisation et calcul de la surprise** — standardisation de chaque indicateur par rapport à sa distribution historique.
3. **Agrégation exponentielle en un score de signal** — pondération des surprises passées avec décroissance temporelle.
4. **Validation empirique du signal** — mesure de la corrélation (IC) et du ratio information (IR) vis-à-vis des rendements forward.

Architecture des données

Principe de non-réplication

Une contrainte fondamentale du modèle est que les données ne sont **jamais recopiées en avant** d'une date de publication à la suivante. Chaque `DataFrame` par indicateur ne contient une valeur qu'aux dates effectives de release :

> *"Les timestamps correspondent uniquement aux dates de release, pas de recopie des données pour garder l'effet de surprise."*

Cette discipline évite d'introduire du look-ahead bias dans le calcul du score global, qui agrège uniquement la *dernière valeur publiée* de chaque indicateur via `.iloc[-1]`.

Structure d'entrée

Le dictionnaire `ZONES_TICKERS` (défini dans `settings.py`) cartographie zones monétaires et indicateurs. La fonction `GetData()` récupère l'ensemble des séries, en séparant soigneusement les données par indicateur pour préserver l'intégrité temporelle.

Calcul du Score : Formules Mathématiques

Normalisation de l'indicateur (Activity Score)

Pour chaque indicateur i à la date t, la valeur normalisée est :

$$A_{i,t} = \frac{\text{Actual}_t - \text{Average}_t}{\sigma_{i,\,\text{rolling}}}$$

où Average_t est la moyenne glissante de l'indicateur et $\sigma_{i,\,\text{rolling}}$ est son écart-type sur une fenêtre historique. Cette normalisation exprime la position courante de l'indicateur par rapport à sa tendance récente.

Score de Surprise

La surprise économique mesure l'écart entre la valeur publiée (*Actual*) et le consensus des analystes (*Forecast*), normalisé par la volatilité historique de l'indicateur :

$$S_{i,t} = \frac{\text{Actual}_t - \text{Forecast}_t}{\sigma_{i,\,\text{rolling}}}$$

Un $S_{i,t} > 0$ signifie que la publication a dépassé les attentes (surprise positive) ; $S_{i,t} < 0$ signifie une déception.

Signal Agrégé avec Décroissance Temporelle

Les surprises passées sont combinées en un signal continu grâce à une somme pondérée exponentiellement, paramétrée par un demi-vie λ :

$$\text{Signal}_i(t) = \sum_{k\,:\,t_k<t} w_{i,t_k} \cdot S_{i,t_k} \cdot e^{-\lambda(t-t_k)}$$

Le terme $e^{-\lambda(t-t_k)}$ assure que les surprises récentes pèsent davantage que les surprises anciennes, simulant la *mémoire décroissante* du marché vis-à-vis d'une publication. Les poids w_{i,t_k} permettent une pondération supplémentaire par indicateur (définie dans `weights_finder.ipynb`).

Score Final par Zone

Le score global d'une zone monétaire est la somme pondérée des signaux individuels de chaque indicateur j appartenant à cette zone :

$$\text{ScoreFinal} = \sum_j S_{\text{ind},j} \times w_{\text{ind},j}$$

Ce score synthétise en un chiffre l'état macroéconomique relatif de la zone : positif implique une dynamique plus forte que prévu, négatif une déception généralisée.

Exemple d'output (données en production)

Devise	Inflation	Emploi	Taux	Score Global
USD	1.31	–0.44	–6.56	–5.69
EUR	0.56	0.00	–1.63	–1.07
JPY	0.00	–1.69	5.95	4.26
GBP	0.01	–2.74	–4.35	–7.09
CAD	–0.00	1.91	–3.63	–1.72
AUD	0.33	–0.13	–3.77	–3.56

Implémentation Python — Points Clés

Initialisation et récupération des données

Listing 1: Chargement des modules et collecte des données

```python
from files.engine import GetWeights, ComputeScore
from files.data_fetch import GetData
from settings import *

# Recuperation des donnees macro pour toutes les zones
data = GetData(ZONES_TICKERS, fred_api)
```

La fonction `GetData` interroge l'API FRED ainsi que d'autres sources configurées dans `settings.py`. Le résultat est un dictionnaire structuré par zone et par indicateur, respectant strictement le principe de non-réplication temporelle.

Calcul du score et historique

Listing 2: Calcul du score et export de l'historique

```python
# Calcul du score courant et de l'historique journalier
df_final, df_history = ComputeScore(data_map=data)

# Export CSV pour analyses downstream
df_history.to_csv("df_score.csv")
```

`ComputeScore` retourne deux objets :

- `df_final` — snapshot courant du score par zone et par catégorie.
- `df_history` — série temporelle du score global par devise, permettant l'analyse de momentum.

Validation : Information Coefficient et Information Ratio

Listing 3: Fonction d'analyse du signal (style Prado)

```python
from scipy.stats import spearmanr
```

```python

def analyze_signal(signal, target, window=20):
    df_calc = pd.concat([signal, target], axis=1).dropna()
    s, t = df_calc.iloc[:, 0].values, df_calc.iloc[:, 1].values

    # IC global (Spearman rank correlation)
    ic, p_val = spearmanr(s, t)

    # IC glissant sur fenetre de 20 jours
    rolling_ic = []
    for i in range(window - 1, len(s)):
        val, _ = spearmanr(s[i-window+1:i+1], t[i-window+1:i+1])
        rolling_ic.append(val)

    ic_mean = np.nanmean(rolling_ic)
    ic_std  = np.nanstd(rolling_ic)
    ir = ic_mean / ic_std if ic_std != 0 else 0

    return ic, p_val, ir, rolling_ic
```

Métriques de Validation du Signal

Information Coefficient (IC)

L'IC est calculé comme la corrélation de rang de Spearman entre le signal macro et les *rendements forward* à $t+1$ du marché correspondant. La corrélation de Spearman est préférée à Pearson car elle est insensible aux valeurs extrêmes et capture les relations monotones non-linéaires.

$$\text{IC} = \rho_s(\text{Signal}_t,\ \text{Ret}_{t+1})$$

Un IC positif indique que le signal prédit correctement le signe des rendements futurs. Les seuils empiriques de l'industrie considèrent :

- $|\text{IC}| > 0.05$: signal exploitable,
- $|\text{IC}| > 0.10$: signal fort.

Information Ratio (IR)

L'IR mesure la *stabilité* de l'IC dans le temps, en divisant sa moyenne par son écart-type sur une fenêtre glissante :

$$\text{IR} = \frac{\overline{\text{IC}}_{\text{rolling}}}{\sigma_{\text{IC rolling}}}$$

Un IR élevé (typiquement > 0.5) indique que la prédictivité du signal est constante plutôt qu'épisodique, ce qui est crucial pour la robustesse opérationnelle d'un modèle de trading.

Résultats en Production (Exemple)

Paire	IC	P-Value	IR
USD vs DXY	0.00672	0.881	0.030
EUR vs EURUSD	–0.01316	0.770	–0.231

Les valeurs d'IC proches de zéro et des p-values élevées indiquent que sur la période testée, le signal ne présente pas de valeur prédictive statistiquement significative. Ce résultat appelle à deux types d'investigation : extension de la fenêtre historique et/ou révision du mapping signal → actif.

Pipeline Complet : Flux de Données

`FRED API / Sources externes`

↓

`GetData()` — *parsing, nettoyage, alignement temporel*

↓

`ComputeScore()` — *normalisation, surprise, décroissance exponentielle*

↓

`df_final` + `df_history`

↓

Construction des signaux & rendements forward

↓

`analyze_signal()` — *IC Spearman, Rolling IC, IR*

↓

Visualisation & Export

Limites et Pistes d'Amélioration

Limites actuelles identifiées :

- La fenêtre de données disponible est courte (quelques semaines visibles dans l'historique), ce qui fragilise statistiquement les IC calculés.
- Le mapping signal → actif est direct (score USD → DXY) sans prise en compte des effets croisés entre zones.
- La p-value élevée signale un manque de puissance statistique plutôt qu'une absence réelle de signal.

Pistes d'amélioration :

- Étendre l'historique de backtest pour augmenter la puissance statistique.

- Tester un signal différentiel (score USD moins score EUR) directement contre EURUSD plutôt que les scores individuels.
- Introduire un IC bootstrappé pour quantifier l'incertitude d'estimation.
- Calibrer les demi-vies λ par indicateur et par régime de volatilité.

Glossaire

IC *Information Coefficient* — corrélation rang entre signal et rendement forward.

IR *Information Ratio* — moyenne de l'IC glissant divisée par son écart-type.

Surprise Écart normalisé entre la valeur publiée et le consensus (Forecast).

Half-life (λ)
Paramètre de décroissance exponentielle contrôlant la mémoire du signal.

Look-ahead bias
Biais introduit quand une donnée future est utilisée pour calculer un signal passé.

Forward return
Rendement de l'actif à $t+1$, utilisé comme cible de prédiction.