

ІДЗ — індивідуальне дослідницьке завдання

Задача: Розробка адаптивної системи вибору інвестиційної стратегії на основі динамічної якості даних

Мета: Розробити програмне забезпечення на Python, що дозволяє досліджувати вплив якості вхідних фінансових даних та аномалій на кінцевий рейтинг активів, побудований за методом Сааті. Данні отримуються з Yahoo Finance.

Постановка задачі

Необхідно розробити програмну систему, яка:

1. Отримує фінансові дані компаній із зовнішнього джерела
2. Формує критерії оцінки активів
3. Використовує метод аналізу ієрархій (АНР, метод Сааті) для ранжування активів
4. Моделює погіршення якості даних
5. Досліджує вплив якості даних на:
 - стабільність ранжування
 - узгодженість матриці попарних порівнянь (CR)
6. Реалізує адаптивний механізм корекції ваг критеріїв залежно від якості даних

Архітектура системи

Рекомендується розбити систему на модулі:

1. Data Ingestion (ETL — Extract)

Джерело: Yahoo Finance

Інструмент: Python-бібліотека «yfinance»

Вимоги:

- 3–9 компаній (тикери задаються користувачем)
- період: останні 5 років
- частота: daily / weekly

Вихід: DataFrame з полями Date, Close (Price), Volume, інші (опціонально)

2. Feature Engineering (Transform)

Побудувати критерії для АНР, наприклад:

- Середня дохідність (Return)
- Волатильність (Risk)
- Обсяг торгів (Liquidity)
- Максимальна просадка (Drawdown)

Нормалізація: Min-Max або Z-score

3. Реалізація АНР (метод Сааті)

Основні етапи:

1. Формування матриці попарних порівнянь критеріїв
2. Обчислення власного вектора (ваг)
3. Обчислення:
 - λ_{\max}
 - CI (Consistency Index)
 - CR (Consistency Ratio)

Умова:

$CR \leq 0.1$ — допустима узгодженість

4. Модуль деградації даних

Реалізувати генератор шуму:

Типи шуму:

1. Пропуски (NaN)
2. Викиди (outliers): множення значень на коефіцієнт (наприклад $\times 2$ – $\times 5$)
3. Затримки (lag): зсув значень у часі

Параметри:

- рівень деградації: 5%, 10%, 15%, 20%, 25%, 30%
- деградується лише один критерій

Експериментальна частина

1. Для кожного рівня шуму:

- Перерахувати критерії
- Побудувати АНР
- Обчислити:
 - CR
 - рейтинг активів

Повторити експеримент 5 разів (через випадковість шуму)

2. Проаналізувати стійкість

Побудувати залежності:

1. CR vs Noise Level
2. Зміна позицій активів (rank shift)

Визначити поріг шуму, при якому CR > 0.1, рейтинг стає нестабільним.

3. Адаптивний "фільтр якості"

Ввести метрику якості даних. Наприклад: % пропусків, дисперсія, кількість викидів.

Правило:

if quality < threshold:

weight_i = weight_i * alpha (alpha < 1)

Варіанти:

- жорстке відключення критерію
- плавне зменшення ваги

Очікувані результати

1. Реалізація системи:

- Програмний код розробленої системи
- Пояснення та коментарі
- Графічне представлення реалізованої архітектури

2. Графіки:

- CR vs % шуму
- зміна рангу активів
- вплив різних типів шуму

3. Висновки