

Modele
python de

```
28
29 # 3. Volatilité avec EWMA
30 def ewma_volatility(returns, lambda_=0.94):
31     ewma_vol = returns.ewm(span=(2/(1-lambda_)-1), adjust=False).std()
32     return ewma_vol[-1]
33
34 # 4. Volatilité avec Random Forest
35 def rf_volatility(returns, n_lags=5):
36     X = np.array([returns[i-n_lags:i] for i in range(n_lags, len(returns))])
37     y = returns[n_lags:]
38
39     scaler = StandardScaler()
40     X_scaled = scaler.fit_transform(X)
41
42     X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, shuffle=False)
43
44     model = RandomForestRegressor(n_estimators=100, random_state=42)
45     model.fit(X_train, y_train)
46
47     y_pred = model.predict(X_test)
48     rmse = np.sqrt(mean_squared_error(y_test, y_pred))
49     return rmse
50
51 # 5. Evaluation de la pertinence des modèles (erreur RMSE et AIC/BIC)
52 def evaluate_model_performance(model_volatility, actual_volatility, model_name, model_type=""):
53     rmse = np.sqrt(np.mean((model_volatility - actual_volatility)**2))
54     print(f"RMSE de {model_name}: {rmse:.4f}")
55     return rmse
56
57 def calculate_weights(pertinence_scores):
58     total_score = sum(pertinence_scores.values())
59     weights = {model_name: score / total_score for model_name, score in pertinence_scores.items()}
60     return weights
```

quantification de risque/opportunité :

Lorsque le code sera terminé le lien vers le repository github figurera ici : [lien](#)

Calcul de la Volatilité

Cinq branches pour le calcul de volatilité

Chaque branche utilise un modèle distinct pour évaluer la volatilité. Lors du vote final, la pertinence de chaque branche sera déterminée et intégrée via un coefficient de pondération.

Les branches proposées :

GARCH (Generalized Autoregressive Conditional Heteroskedasticity) :

- Modèle bien adapté aux séries temporelles financières pour modéliser les comportements volatils.

LSTM (Long Short-Term Memory) :

- Réseau neuronal récurrent pour capturer des schémas non linéaires dans les séries temporelles.

Écart-type des rendements historiques :

- Méthode classique, simple et rapide.

EWMA (Exponentially Weighted Moving Average) :

- Prend en compte un poids décroissant sur les données plus anciennes.

Random Forest pour prédire la volatilité future :

- Combine plusieurs arbres de décision pour une meilleure robustesse.

Pertinence des branches

Pour déterminer la pertinence de chaque branche, on peut :

Évaluer la précision historique :

- Comparer les prédictions avec la volatilité réalisée (ex-post) sur une période donnée.
- Calculer un score de RMSE (Root Mean Square Error).

Adaptabilité au contexte de marché :

- Détecter les régimes de marché (volatilité basse/modérée/élevée) et attribuer un poids supérieur aux modèles performants dans ces régimes.

Corrélation avec les événements :

- Mesurer si le modèle capture efficacement l'impact des nouvelles ou événements externes sur la volatilité.

Rapport de Volatilité du Portefeuille

Volatilités par Modèle

Modèle	Volatilité	Score Pertinence
EGARCH	0.0005	0.0071
Random Forest	0.0106	0.0010
Historique	0.0097	-0.0014
EWMA	0.0089	-0.0007

Volatilité Totale du Portefeuille

0.0069

Poids des Modèles dans la Décision

Modèle	Poids
EGARCH	1.1589
Random Forest	0.1715
Historique	-0.2232
EWMA	-0.1072

Vote final

Pondération dynamique des branches en fonction de leur pertinence calculée.

Calcul final de la volatilité pour chaque actif séparément.

Calcul de la volatilité totale du portefeuille :

$$\text{Volatilité Portefeuille} = \sqrt{\sum_i w_i^2 \cdot \sigma_i^2 + \sum_{i \neq j} w_i \cdot w_j \cdot \sigma_i \cdot \sigma_j \cdot \rho_{ij}}$$

$$\text{Volatilité Portefeuille} = \sqrt{\sum_i w_i^2 \cdot \sigma_i^2 + 2 \sum_{i < j} w_i \cdot w_j \cdot \sigma_i \cdot \sigma_j \cdot \rho_{ij}}$$

- w_i : poids de l'actif i ,
- σ_i : volatilité de l'actif i ,
- ρ_{ij} : corrélation entre les actifs i et j .

Affichage HTML

Pour chaque actif : Afficher sa volatilité calculée.

Pour le portefeuille : Afficher la volatilité totale pondérée.

Qualification du Risque/Opportunité

Les branches proposées

Branche 1 : Calcul de la profitabilité

Basé sur les fondamentaux économiques et financiers du module ai-hedge-fund/src/agents/fundamentals.py.

Produit une note entre -1 (risque majeur) et 1 (opportunité forte).

Branche 2 : Analyse de sentiment général

Modèles NLP (ex : BERT ou GPT) pour analyser les actualités et la perception du marché vis-à-vis de chaque actif.

Intégration des données textuelles issues d'APIs.

Branches 3 à 5 : Prompts pour les actualités

Fonctionnement :

- Générer un prompt spécifique pour chaque actif, demandant une analyse des actualités politiques, géopolitiques, économiques et financières.
- Exemple de prompt : *"Analyse les actualités politiques et économiques pour [nom actif]. Donne une note entre -1 et 1 pour représenter le risque ou l'opportunité."*
- L'utilisateur interagit avec ChatGPT, puis injecte les réponses via une zone de texte.

Vote final

Chaque branche produit une note.

Pondération des branches selon leur précision ou fiabilité.

Calcul d'une note finale pour chaque actif.

Interrelations entre actifs

Construction du réseau d'interrelations

Méthode :

- Utiliser des modèles VAR (Vector AutoRegression) pour capturer les dépendances dynamiques.
- Estimation des effets directionnels (positifs/négatifs).

Quantification des relations :

- Score de dépendance entre -1 (effet inverse) et 1 (effet direct positif).

Impact sur le portefeuille :

- Ajustement des notes de risque/opportunité en fonction des relations systémiques.

Calcul du Score Final du Portefeuille

Formule principale :

Score Final = $\sum w_i \cdot (\text{Volatilité}_i \times \text{Risque} / \text{Opportunité}_i)$
 $\text{Score Final} = \sum w_i \cdot (\text{Volatilité}_i \times \text{Risque} / \text{Opportunité}_i)$

Prise en compte des relations :

- Ajuster le score final pour refléter les interdépendances.

Optimisation et Robustesse

Optimisation

Utiliser la théorie moderne du portefeuille (Markowitz) pour optimiser les pondérations :

Minimiser le risque pour un rendement attendu donné.
 Minimiser le risque pour un rendement attendu donné.
 Minimiser le risque pour un rendement attendu donné.

Robustesse

Stress Testing :

- Simuler des scénarios défavorables (crises, hausses soudaines).

Interprétabilité :

- Utiliser SHAP pour expliquer les décisions des modèles.