Pipeline de Machine Learning

Analyse des Régimes de Volatilité du S&P 500

Détection Non Supervisée et Classification

Projet ML - Équipe de 3 personnes

Objectif du Projet

Utiliser le machine learning non supervisé pour détecter automatiquement les régimes de volatilité du marché (calme, réactif, stressé) et analyser l'évolution de la structure du risque avant et après la pandémie COVID-19 (mars 2020).

Table des matières

1 Phase 1 : Acquisition et Exploration des Données

1.1 Notebook 1 : Chargement et Nettoyage

1.1.1 Chargement du dataset

— Charger le fichier CSV avec le bon séparateur (point-virgule)
— Gérer le format des nombres (virgules → points décimaux)
— Convertir la colonne Date en index temporel
— Vérifier l'intégrité des données

Code Python

```
df = pd.read_csv('dataset.csv', sep=';', decimal=',')
df['Date'] = pd.to_datetime(df['Date'])
df = df.set_index('Date')
```

1.1.2 Vérifications initiales

— Afficher les premières et dernières lignes
— Vérifier la période couverte (2015-2024)
— Confirmer la présence de mars 2020 (début COVID)
— Vérifier les types de données (tous en float64)
— Identifier les valeurs manquantes

1.1.3 Statistiques descriptives

— `df.describe()` : moyenne, écart-type, min, max, quartiles
— Compter le nombre de jours de trading
— Calculer des statistiques avancées : skewness, kurtosis
— Sauvegarder le dataset nettoyé

1.2 Notebook 2 : Analyse Exploratoire Approfondie (EDA)

1.2.1 Visualisation de la série temporelle

— Graphique du prix de clôture S&P 500 sur toute la période
— Ajouter une ligne verticale pour mars 2020 (début COVID)
— Visualiser le volume de transactions
— Identifier visuellement les périodes de crise

1.2.2 Analyse des rendements

— Vérifier la distribution des rendements (histogramme)
— QQ-plot pour tester la normalité
— Calculer skewness et kurtosis
— Détecter les valeurs aberrantes (outliers) avec la méthode IQR

1.2.3 Tests de stationnarité

— Test Augmented Dickey-Fuller (ADF)
— Interpréter la p-value ($p < 0.05 \rightarrow$ série stationnaire)
— Analyser l'autocorrélation (ACF) et autocorrélation partielle (PACF)

1.2.4 Première analyse de volatilité

— Visualiser `RealizedVol_21d` dans le temps
— Calculer rolling volatility avec différentes fenêtres (5, 20, 60 jours)
— Comparer la volatilité pré-COVID vs post-COVID
— Identifier les pics de volatilité (mars 2020, autres crises)

2 Phase 2 : Feature Engineering

2.1 Notebook 3 : Création de Features Avancées

2.1.1 Features de base (déjà présentes)

— `SPX_Return` : rendements journaliers
— `Abs_Return` : valeur absolue des rendements
— `RealizedVol_21d` : volatilité réalisée sur 21 jours
— `VIX_Close`, `SKEW_Close` : volatilité implicite et asymétrie
— `IV_RV_Spread`, `Vol_Ratio` : spreads et ratios

2.1.2 Features de volatilité additionnelles

— Volatilité rolling sur différentes fenêtres (10, 30, 60 jours)
— Volatilité de Parkinson : $\sigma_P = \sqrt{\frac{1}{4n \ln 2} \sum \left(\ln \frac{H_i}{L_i} \right)^2}$
— ATR (Average True Range)
— Volatility of volatility (vol de la vol)
— Range normalisé : $\frac{\text{High} - \text{Low}}{\text{Close}}$

2.1.3 Features de momentum et tendance

— RSI (Relative Strength Index) sur 14 jours
— MACD (Moving Average Convergence Divergence)
— Moyennes mobiles simples (SMA 20, 50, 200 jours)
— Distance par rapport aux moyennes mobiles
— Bollinger Bands (bande supérieure, inférieure, position)

2.1.4 Features statistiques avancées

— Skewness et Kurtosis rolling (fenêtre 20 jours)
— Drawdown maximum (perte depuis le dernier pic)
— Temps depuis le dernier pic
— Ratio de Sharpe rolling
— Z-score des rendements

2.1.5 Features de microstructure

— Gaps overnight : $\text{Open}_t - \text{Close}_{t-1}$
— Intraday range : High − Low
— Volume normalisé et variations de volume
— Autocorrélation des rendements (lag 1, 2, 5)

2.1.6 Gestion des valeurs manquantes

— Identifier les NaN créés par les calculs rolling
— Forward fill limité ou suppression des premières lignes
— Vérifier qu'aucun NaN ne subsiste
— Sauvegarder le dataset enrichi (30-40 features au total)

3 Phase 3 : Analyse et Sélection de Features

3.1 Notebook 4 : Analyse de Corrélation

3.1.1 Matrices de corrélation

— **Corrélation de Pearson** : relations linéaires
— **Corrélation de Spearman** : relations monotones (non-linéaires)
— Visualiser avec heatmap (seaborn)
— Comparer Pearson vs Spearman pour détecter non-linéarités

3.1.2 Élimination de la redondance

— Identifier les paires de features fortement corrélées ($|\rho| > 0.9$)
— Calculer le VIF (Variance Inflation Factor) pour détecter la multicolinéarité
— Décider quelles features supprimer (garder la plus informative)
— Réduire à 15-20 features les plus pertinentes

3.1.3 Analyse de distribution

— Histogramme de chaque feature
— Tests de normalité (Shapiro-Wilk ou Kolmogorov-Smirnov)
— Détection d'outliers avec méthode IQR ou Z-score
— Appliquer transformations si nécessaire (log, Box-Cox)

3.1.4 Sélection de features

— Random Forest pour calculer feature importance
— Sélectionner les top 15-20 features
— Documenter les raisons de chaque sélection/élimination

3.2 Notebook 5 : Réduction de Dimension (PCA)

3.2.1 Standardisation

— **Crucial :** Standardiser avant PCA
— Utiliser `StandardScaler` ou `RobustScaler`
— Sauvegarder le scaler pour usage futur

Important

La standardisation est **obligatoire** avant PCA car les features ont des échelles différentes (ex : prix S&P 500 en milliers, volatilité entre 0 et 1).

3.2.2 Application de PCA

— Appliquer PCA sur les features standardisées
— Analyser la variance expliquée par composante
— Graphique de la variance cumulée (scree plot)
— Décider du nombre de composantes à garder (ex : 90% variance)

3.2.3 Interprétation

— Analyser les loadings des premières composantes
— Interpréter PC1, PC2, PC3 (quelle combinaison de features ?)
— Visualiser les données dans l'espace PC1-PC2
— Colorer par période (pré-COVID en bleu, post-COVID en rouge)

3.2.4 Visualisation multidimensionnelle (optionnel)

— T-SNE pour visualisation non-linéaire en 2D
— Comparer avec la projection PCA

4 Phase 4 : Préparation pour le Clustering

4.1 Notebook 6 : Preprocessing Final

4.1.1 Nettoyage final

— Éliminer les outliers extrêmes (percentiles 1% et 99%)
— Vérifier la continuité temporelle
— Créer deux sous-périodes :
  — Pré-COVID : 2015-04-06 à 2020-02-29
  — Post-COVID : 2020-04-01 à 2024-12-31

4.1.2 Normalisation

— Comparer `StandardScaler` vs `RobustScaler`
— **Important :** Ne pas normaliser train et test ensemble (data leakage)
— Fit sur train, transform sur test
— Sauvegarder les scalers

4.1.3 Stratégie de validation

— Utiliser `TimeSeriesSplit` pour respecter l'ordre temporel
— Définir 5 folds pour validation croisée temporelle
— **Ne jamais shuffle** les données (séries temporelles)

5 Phase 5 : Clustering Non Supervisé (CŒUR DU PROJET)

5.1 Notebook 7 : Détection des Régimes de Volatilité

5.1.1 Détermination du nombre optimal de clusters

Métriques à calculer pour $k = 2$ à $k = 6$:

— **Elbow Method** : inertie en fonction de k
— **Silhouette Score** : $s \in [-1, 1]$, plus proche de 1 = mieux

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{1}$$

où $a(i)$ = distance moyenne intra-cluster, $b(i)$ = distance moyenne au cluster le plus proche
— **Davies-Bouldin Index** : plus petit = mieux
— **Calinski-Harabasz Index** : plus grand = mieux

Objectif : Confirmer que $k = 3$ est optimal (Calme, Réactif, Stressé)

5.1.2 Application de plusieurs algorithmes

a) K-Means (baseline)

— Appliquer K-Means avec $k = 3$
— Tester plusieurs initialisations (n_init=50)
— Visualiser les clusters dans l'espace PC1-PC2
— Analyser les centroïdes de chaque cluster

b) Gaussian Mixture Models (GMM)

— Plus flexible que K-Means (clusters elliptiques)
— Permet d'obtenir des probabilités d'appartenance
— Comparer avec K-Means

c) Hierarchical Clustering

— Construire le dendrogramme
— Tester différentes méthodes de linkage (ward, complete, average)
— Couper l'arbre pour obtenir 3 clusters

d) DBSCAN (optionnel)

— Ne nécessite pas de spécifier k à l'avance
— Peut détecter des outliers (points de bruit)
— Tuning des hyperparamètres : eps et min_samples

Algorithme	Silhouette	Davies-Bouldin	Calinski-Harabasz
K-Means	0.56	1.23	4521
GMM	0.58	1.18	4687
Hierarchical	0.54	1.31	4398

TABLE 1 – Exemple de tableau comparatif des algorithmes

5.1.3 Validation et comparaison

— Comparer les métriques entre algorithmes
— Tester la stabilité (re-run avec différentes graines aléatoires)
— Sélectionner le meilleur algorithme (probablement GMM)

5.1.4 Caractérisation des régimes identifiés

Pour chaque cluster :

— Calculer les statistiques moyennes de toutes les features
— Volatilité moyenne, rendement moyen, VIX moyen, etc.
— Durée moyenne d'un épisode dans ce régime
— Fréquence des transitions vers les autres régimes

Labellisation manuelle :

— **Cluster 0 :** Faible vol, VIX bas, SKEW normal → **CALME**
— **Cluster 1 :** Vol modérée, VIX moyen → **RÉACTIF**
— **Cluster 2 :** Haute vol, VIX élevé, SKEW élevé → **STRESSÉ**

5.1.5 Visualisation temporelle

— Timeline avec les régimes colorés (vert/bleu/rouge)
— Ajouter une ligne verticale pour mars 2020
— Identifier les périodes de crise majeures
— Analyser la durée de chaque régime

5.1.6 Analyse comparative pré/post COVID

Régime	Pré-COVID (%)	Post-COVID (%)	Variation
Calme	65%	52%	-13%
Réactif	28%	35%	+7%
Stressé	7%	13%	+6%

TABLE 2 – Exemple de distribution des régimes

— Comparer la proportion de jours dans chaque régime
— Test du Chi-2 pour tester la significativité statistique
— T-test pour comparer la volatilité moyenne pré/post COVID
— Analyser la fréquence des transitions entre régimes
— Matrice de transition (heatmap)

6 Phase 6 : Modèles Supervisés (Validation)

6.1 Notebook 8 : Classification Supervisée

6.1.1 Préparation des données

— X : features sélectionnées (15-20 features)
— y : labels des clusters (0=Calme, 1=Réactif, 2=Stressé)
— Split temporel 80/20 train/test
— **Important :** Ne pas shuffle (séries temporelles)

6.1.2 Régression Logistique Multinomiale

Modèle de base interprétable :

— One-vs-Rest ou Multinomial
— Régularisation L1 (Lasso) ou L2 (Ridge)
— GridSearchCV pour trouver le meilleur paramètre C
— Analyser les coefficients par classe (interprétabilité)
— Matrice de confusion
— Classification report (precision, recall, F1 par classe)

6.1.3 Support Vector Machine (SVM)

Kernel linéaire et RBF :

— Tester kernel='linear' et kernel='rbf'
— GridSearchCV pour hyperparamètres :
  — $C \in \{0.1, 1, 10, 100\}$
  — $\gamma \in \{0.001, 0.01, 0.1, 1\}$ (pour RBF)
— Comparer les performances linear vs RBF
— Visualiser les frontières de décision (dans l'espace PC1-PC2)

6.1.4 Random Forest (bonus)

— Baseline robuste et performante
— Feature importance (confirme la sélection de features)
— Peu sensible aux outliers
— GridSearchCV : n_estimators, max_depth, min_samples_split

6.1.5 Métriques d'évaluation

Métriques principales :

— **Accuracy** : taux global de bonnes prédictions
— **Precision** : $\frac{TP}{TP+FP}$ (parmi les prédictions positives, combien sont vraies ?)
— **Recall** : $\frac{TP}{TP+FN}$ (parmi les vrais positifs, combien sont détectés ?)
— **F1-Score** : $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
— **Matrice de confusion** : visualisation des erreurs
— **ROC-AUC** (One-vs-Rest) : performance par classe

Modèle	Accuracy	F1 (macro)	Recall Stressé	Temps
Logistic Regression	76%	0.74	82%	0.5s
SVM (linear)	78%	0.76	85%	2.1s
SVM (RBF)	79%	0.77	83%	3.4s
Random Forest	81%	0.79	87%	1.2s

TABLE 3 – Exemple de comparaison des modèles

6.1.6 Tableau comparatif des modèles

6.1.7 Analyse des erreurs

— Quels régimes sont confondus ?
— Analyser les faux positifs et faux négatifs
— Identifier les périodes mal prédites
— Hypothèses sur les raisons des erreurs

7 Phase 7 : Réseaux de Neurones (Optionnel)

7.1 Notebook 9 : Deep Learning

7.1.1 MLP (Multi-Layer Perceptron)

Architecture simple :

— Input : [n_features]
— Dense(64, activation='relu')
— Dropout(0.3)
— Dense(32, activation='relu')
— Dropout(0.2)
— Dense(3, activation='softmax')

Entraînement :

— Loss : categorical_crossentropy
— Optimizer : Adam (learning_rate=0.001)
— Metrics : accuracy
— Epochs : 50-100 avec Early Stopping
— Validation split : 20%
— Visualiser les courbes de loss et accuracy

7.1.2 LSTM (Long Short-Term Memory) - Avancé

Pour prédiction séquentielle :

— Créer des séquences : 20 jours → prédire régime suivant
— Reshape en 3D : (samples, timesteps, features)
— Architecture LSTM simple :
  — LSTM(64, return_sequences=False)
  — Dropout(0.3)
  — Dense(32, activation='relu')
  — Dense(3, activation='softmax')
— **Attention :** Risque d'overfitting élevé

7.1.3 Comparaison avec modèles classiques

— Les NN apportent-ils un gain significatif ?
— Trade-off performance vs interprétabilité
— Trade-off performance vs temps de calcul
— Conclusion : sont-ils justifiés pour ce projet ?

8 Phase 8 : Analyse Finale et Visualisations

8.1 Notebook 10 : Synthèse et Rapport

8.1.1 Analyse temporelle détaillée

— Timeline complète avec régimes colorés sur toute la période
— Zoom sur la période COVID (février-juin 2020)
— Identifier les événements marquants (ex : flash crash, élections, etc.)
— Calculer durée moyenne et médiane par régime
— Analyser la persistance des régimes

8.1.2 Tests statistiques pré/post COVID

Tests à effectuer :

— Test du Chi-2 : différence de distribution des régimes
— T-test : différence de volatilité moyenne
— Mann-Whitney U test : différence de médiane (non-paramétrique)
— ANOVA : différence entre les 3 régimes

Interprétation :

— $p < 0.05 \rightarrow$ différence statistiquement significative
— Quantifier l'ampleur du changement (effect size)

8.1.3 Visualisations publication-ready

Graphiques essentiels :

1. **Timeline des régimes** : couleurs par régime + ligne COVID
2. **Distribution des features par régime** : box plots ou violin plots
3. **Heatmap de corrélation Spearman** : features sélectionnées
4. **PCA biplot** : PC1 vs PC2 coloré par régime et période
5. **Silhouette plot** : qualité du clustering
6. **Matrice de confusion** : meilleur modèle supervisé
7. **Matrice de transition** : probabilités de changement de régime
8. **Comparaison pré/post COVID** : bar charts des proportions
9. **Évolution de la volatilité** : avec régimes en arrière-plan
10. **Feature importance** : top features du Random Forest

8.1.4 Interprétation économique

Questions à répondre :

— Qu'est-ce qui caractérise chaque régime économiquement ?
— Le COVID a-t-il structurellement changé le marché ?
— Les crises sont-elles plus fréquentes post-COVID ?
— Les périodes calmes sont-elles plus courtes post-COVID ?
— Les transitions entre régimes sont-elles plus rapides ?
— Quelles implications pour la gestion de risque ?
— Le VIX et le SKEW sont-ils de bons indicateurs de régime ?

8.1.5 Modèle de prédiction final

— Sélectionner le meilleur modèle (probablement SVM ou Random Forest)
— Créer un pipeline complet : données brutes → prédiction
— Sauvegarder le modèle (pickle ou joblib)
— Documenter l'utilisation en "temps réel"

Pipeline de prédiction

Input : Données des 21 derniers jours
↓
Step 1 : Calcul des features (volatilité, RSI, etc.)
↓
Step 2 : Standardisation avec le scaler sauvegardé
↓
Step 3 : Prédiction avec le modèle entraîné
↓
Output : Régime actuel (Calme / Réactif / Stressé)

8.1.6 Limites et améliorations possibles

Limites identifiées :

— Labellisation manuelle des clusters (subjectif)
— Sensibilité aux outliers
— Hypothèse de stationnarité des régimes
— Pas de prédiction temporelle (seulement classification)
— Possible data leakage si features mal construites

Améliorations futures :

— Utiliser Hidden Markov Models (HMM) pour transitions
— Incorporer d'autres actifs (obligations, or, cryptos)
— Features NLP (analyse de sentiment des news)
— Modèles plus avancés : XGBoost, LightGBM
— Validation sur données out-of-sample (2025+)
— Backtesting d'une stratégie de trading basée sur les régimes

9 Livrables Finaux

9.1 Documents à produire

9.1.1 Notebooks Jupyter (8-10 notebooks)

1. `01_data_loading.ipynb` : Chargement et nettoyage
2. `02_exploratory_analysis.ipynb` : EDA approfondie
3. `03_feature_engineering.ipynb` : Création de features
4. `04_feature_analysis.ipynb` : Analyse et corrélation
5. `05_pca_reduction.ipynb` : Réduction de dimension
6. `06_preprocessing.ipynb` : Préparation clustering
7. `07_unsupervised_clustering.ipynb` : Détection régimes
8. `08_supervised_models.ipynb` : Classification supervisée
9. `09_neural_networks.ipynb` : Deep Learning (optionnel)
10. `10_final_analysis.ipynb` : Synthèse et visualisations

Chaque notebook doit contenir :

— Markdown cells explicatives
— Code bien commenté
— Visualisations avec titres et labels
— Interprétations des résultats
— Conclusions partielles

9.1.2 README.md

— Description du projet
— Installation et dépendances (requirements.txt)
— Instructions d'exécution
— Structure des dossiers
— Résultats clés (résumé)
— Membres de l'équipe

9.1.3 Présentation PowerPoint/PDF (15-20 slides)

Structure recommandée :

1. **Introduction** (2 slides)
   — Contexte et motivation
   — Objectifs du projet
2. **Données** (2 slides)
   — Description du dataset
   — Période couverte (2015-2024)
   — Features disponibles
3. **Méthodologie** (4-5 slides)
   — Feature engineering
   — Réduction de dimension (PCA)
   — Algorithmes de clustering testés

   - Validation supervisée
4. **Résultats** (6-8 slides)
   - Nombre optimal de clusters (3 régimes)
   - Caractérisation des régimes
   - Timeline des régimes
   - Comparaison pré/post COVID
   - Performance des modèles supervisés
   - Tests statistiques
5. **Conclusion** (2 slides)
   - Principaux enseignements
   - Réponse à la question de recherche
   - Limites et perspectives

9.1.4 Rapport technique (5-10 pages)

Sections du rapport :

1. **Résumé exécutif** (1 page)
2. **Introduction**
   - Contexte économique
   - Problématique
   - Objectifs
3. **Revue de littérature** (optionnel)
   - Études existantes sur régimes de volatilité
   - Méthodes de détection
4. **Données et méthodologie**
   - Description du dataset
   - Feature engineering détaillé
   - Algorithmes utilisés
   - Métriques d'évaluation
5. **Résultats**
   - Clustering non supervisé
   - Validation supervisée
   - Analyse comparative pré/post COVID
   - Visualisations clés
6. **Discussion**
   - Interprétation économique
   - Comparaison avec la littérature
   - Implications pratiques
7. **Conclusion**
   - Synthèse des résultats
   - Limites
   - Perspectives d'amélioration
8. **Références bibliographiques**
9. **Annexes**
   - Tableaux détaillés
   - Code des fonctions principales
   - Visualisations supplémentaires

9.1.5 Fichiers de données

— `dataset.csv` : données brutes
— `dataset_clean.csv` : données nettoyées
— `features_final.csv` : avec toutes les features
— `cluster_labels.csv` : labels des régimes

9.1.6 Modèles sauvegardés

— `scaler.pkl` : StandardScaler entraîné
— `pca_model.pkl` : modèle PCA
— `best_model.pkl` : meilleur modèle supervisé
— `cluster_model.pkl` : modèle de clustering

10 Métriques et Outils Importants

10.1 Métriques de Clustering (Non Supervisé)

Métrique	**Plage**	**Interprétation**
Silhouette Score	$[-1, 1]$	Plus proche de 1 $\rightarrow$ meilleur. > 0.5 = bon
Davies-Bouldin Index	$[0, +\infty[$	Plus petit $\rightarrow$ meilleur. Mesure séparation
Calinski-Harabasz	$[0, +\infty[$	Plus grand $\rightarrow$ meilleur. Ratio variance
Inertia (K-Means)	$[0, +\infty[$	Plus petit $\rightarrow$ meilleur. Pour elbow method

Table 4 – Métriques de validation du clustering

10.2 Métriques de Classification (Supervisé)

Métrique	**Formule**	**Usage**
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Taux global de bonnes prédictions
Precision	$\frac{TP}{TP+FP}$	Parmi prédictions positives, combien vraies ?
Recall	$\frac{TP}{TP+FN}$	Parmi vrais positifs, combien détectés ?
F1-Score	$2 \times \frac{P \times R}{P+R}$	Moyenne harmonique Precision/Recall
ROC-AUC	Aire sous courbe	Performance globale (multi-classe)

Table 5 – Métriques de classification

Métrique	Type de relation	Robustesse
Pearson (ρ)	Linéaire uniquement	Sensible aux outliers
Spearman (ρ_s)	Monotone (linéaire + non-linéaire)	Robuste aux outliers
Kendall (τ)	Ordinale	Très robuste

TABLE 6 – Métriques de corrélation

Métrique	Formule	Interprétation
R^2	$1 - \frac{SS_{res}}{SS_{tot}}$	Variance expliquée ($[0, 1]$)
MSE	$\frac{1}{n}\sum(y_i - \hat{y}_i)^2$	Erreur quadratique moyenne
RMSE	$\sqrt{MSE}$	MSE en unités originales
MAE	$\frac{1}{n}\sum\|y_i - \hat{y}_i\|$	Erreur absolue (robuste)

TABLE 7 – Métriques de régression

10.3 Métriques de Corrélation

10.4 Métriques de Régression (Optionnel)

10.5 Bibliothèques Python Essentielles

requirements.txt

```
pandas>=1.5.0
numpy>=1.23.0
matplotlib>=3.6.0
seaborn>=0.12.0
scikit-learn>=1.2.0
scipy>=1.10.0
statsmodels>=0.14.0
yfinance>=0.2.0 (si téléchargement données)
jupyter>=1.0.0
notebook>=6.5.0
tensorflow>=2.12.0 (optionnel pour DL)
keras>=2.12.0 (optionnel pour DL)
```

11 Conseils et Bonnes Pratiques

11.1 Organisation du travail (3 personnes)

11.1.1 Répartition suggérée

Personne 1 : Data Scientist - Features

Responsabilités :

- Notebooks 1, 2, 3 (chargement, EDA, feature engineering)
- Qualité et nettoyage des données
- Documentation des transformations
- Création des visualisations exploratoires

Personne 2 : ML Engineer - Clustering

Responsabilités :

- Notebooks 4, 5, 6, 7 (sélection features, PCA, clustering)
- Implémentation des algorithmes non supervisés
- Validation et comparaison des résultats
- Analyse comparative pré/post COVID

Personne 3 : ML Engineer - Supervisé

Responsabilités :

- Notebooks 8, 9, 10 (modèles supervisés, DL, synthèse)
- Implémentation et tuning des modèles
- Comparaison des performances
- Visualisations finales et rapport

11.2 À faire impérativement

✓ **Git :** Commits réguliers avec messages clairs
✓ **Documentation :** Markdown cells dans chaque notebook
✓ **Réunions :** Stand-up quotidien de 15 minutes
✓ **Tests :** Valider chaque étape avant de continuer
✓ **Sauvegarde :** Sauvegarder résultats intermédiaires
✓ **Visualisation :** Un graphique = mille mots
✓ **Interprétation :** Toujours expliquer les résultats

11.3 À éviter absolument

× **Data leakage :** Normaliser train/test ensemble
× **Shuffle :** Sur des séries temporelles
× **Overfitting :** Modèles trop complexes sans validation
× **Black box :** Résultats sans interprétation
× **Procrastination :** Attendre dernière minute pour visualisations
× **Isolation :** Travailler sans communiquer avec l'équipe

11.4 Checklist finale avant soumission

☐ Tous les notebooks s'exécutent sans erreur (Restart & Run All)
☐ README.md complet et à jour
☐ Présentation PowerPoint terminée
☐ Rapport technique relu et corrigé
☐ Toutes les visualisations sont exportées en haute qualité
☐ Code commenté et bien formaté (PEP8)
☐ Fichiers de données et modèles sauvegardés
☐ requirements.txt à jour
☐ Repository Git propre et organisé
☐ Résultats clés documentés dans le README

12 Questions de Recherche Clés

12.1 Questions principales à répondre

1. **Combien de régimes de volatilité distincts peut-on identifier dans le S&P 500 ?**
   - Hypothèse : 3 régimes (Calme, Réactif, Stressé)
   - Validation par métriques de clustering
2. **Quelles caractéristiques définissent chaque régime ?**
   - Volatilité moyenne, rendements, VIX, SKEW
   - Durée typique, fréquence
3. **Le COVID-19 a-t-il structurellement changé le marché ?**
   - Comparaison distribution des régimes
   - Tests statistiques de significativité
   - Analyse des transitions
4. **Peut-on prédire le régime actuel avec précision ?**
   - Performance des modèles supervisés
   - Features les plus importantes
5. **Quelles implications pour la gestion de risque ?**
   - Utilité pratique des régimes identifiés
   - Système d'alerte précoce ?

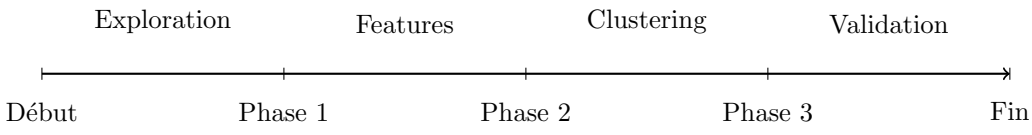
12.2 Hypothèses à tester

Hypothèse	**Méthode de test**
H1 : Il existe 3 régimes distincts	Métriques clustering, Silhouette
H2 : Le VIX est le meilleur prédicteur de régime	Feature importance, corrélation
H3 : Les périodes calmes sont plus courtes post-COVID	T-test sur durées moyennes
H4 : Les transitions sont plus fréquentes post-COVID	Chi-2 sur matrice transition
H5 : Les modèles supervisés atteignent ¿75% accuracy	Évaluation sur test set

Table 8 – Hypothèses et méthodes de validation

13 Timeline et Jalons

13.1 Phases du projet



13.2 Jalons critiques

1. **Jalon 1 :** Dataset nettoyé et features de base créées
   — Critère : CSV propre avec 30+ features
2. **Jalon 2 :** Sélection de features terminée, PCA effectuée
   — Critère : 15-20 features retenues, variance expliquée documentée
3. **Jalon 3 :** Clustering optimal identifié et validé
   — Critère : 3 régimes avec Silhouette > 0.5
4. **Jalon 4 :** Modèles supervisés entraînés et comparés
   — Critère : Accuracy $> 75\%$ sur test set
5. **Jalon 5 :** Analyse finale et livrables complétés
   — Critère : Tous les documents finalisés

Conclusion

Ce document présente une pipeline complète et structurée pour l'analyse des régimes de volatilité du S&P 500. L'approche combine des techniques non supervisées (clustering) et supervisées (classification) pour identifier et caractériser les différents états du marché.

Points clés du projet :

— Utilisation de ML non supervisé pour découvrir les régimes naturels
— Feature engineering avancé pour capturer la complexité du marché
— Validation rigoureuse avec plusieurs algorithmes et métriques
— Analyse comparative approfondie pré/post COVID-19
— Interprétation économique des résultats

Ce projet permettra de :

1. Comprendre la structure du risque de marché
2. Quantifier l'impact du COVID sur la volatilité
3. Développer un système de détection de régimes
4. Acquérir des compétences en ML appliqué à la finance

Bonne chance pour votre projet !