Compte Rendu d'Analyse Prédictive

Modélisation Machine Learning sur Données Financières

ERROUYAS AYA

📋 **Table des Matières**

---

1. Introduction

1.1 Contexte

Dans le secteur financier contemporain, la capacité à prédire avec précision les tendances et les comportements constitue un avantage concurrentiel majeur. L'utilisation de techniques de Machine Learning permet d'extraire des patterns complexes à partir de données historiques, facilitant ainsi la prise de décision stratégique.

Ce projet s'inscrit dans une démarche d'analyse prédictive appliquée à un dataset financier issu de Kaggle (finance-data). Les données comprennent diverses variables numériques et catégorielles représentant des indicateurs financiers potentiellement corrélés à une variable cible d'intérêt.

1.2 Problématique

Question centrale : Comment développer un modèle prédictif robuste capable d'identifier avec précision [la variable cible] à partir des indicateurs financiers disponibles ?

Les défis identifiés incluent :

- La présence potentielle de valeurs manquantes et aberrantes
- La nécessité de gérer des variables catégorielles de cardinalités variables
- L'équilibre entre complexité du modèle et généralisation
- L'optimisation des hyperparamètres dans un espace de recherche étendu

1.3 Objectifs

Objectif principal : Construire un pipeline complet de Machine Learning, depuis le prétraitement jusqu'à l'évaluation rigoureuse de modèles prédictifs.

Objectifs secondaires :

1. Nettoyer et préparer les données selon les meilleures pratiques
2. Explorer les relations entre variables via une analyse exploratoire approfondie
3. Comparer au moins trois algorithmes de Machine Learning différents
4. Optimiser les hyperparamètres via des techniques de recherche systématique
5. Évaluer la performance avec des métriques appropriées et une validation croisée

---

2. Méthodologie

2.1 Stratégie de Prétraitement

2.1.1 Nettoyage des Données

Gestion des doublons

- **Justification** : Les doublons faussent les statistiques et peuvent induire un sur-apprentissage en créant une redondance artificielle dans les données d'entraînement.
- **Approche adoptée** : Détection systématique avec duplicated() et suppression via drop_duplicates().
- **Impact** : Garantit que chaque observation est unique, préservant l'intégrité statistique du dataset.

Formatage et typage

- **Justification** : Les types de données incorrects (ex: nombres stockés comme chaînes) empêchent les opérations mathématiques et les analyses statistiques.
- **Approche adoptée** : Vérification avec info() et conversion explicite si nécessaire.

2.1.2 Imputation des Valeurs Manquantes

Stratégie avancée : KNN Imputer

- **Justification** : Contrairement aux méthodes simples (moyenne, médiane), KNN Imputer exploite les relations entre variables pour estimer les valeurs manquantes. Les k observations les plus similaires sont identifiées dans l'espace des features, et leurs valeurs sont moyennées avec une pondération par distance.
- **Paramètre choisi** : k=5 voisins avec pondération par distance inverse

- **Avantages** :
    - Préserve les corrélations inter-variables
    - Adapté aux données financières où les indicateurs sont souvent corrélés
    - Plus robuste que l'imputation univariée

Variables catégorielles

- **Justification** : Le mode (valeur la plus fréquente) est statistiquement approprié pour les variables discrètes.
- **Approche** : Imputation par modalité majoritaire, avec création d'une catégorie "Unknown" si nécessaire.

2.1.3 Encodage des Variables Catégorielles

Stratégie multi-niveaux basée sur la cardinalité

1. **Label Encoding (cardinalité = 2)**
    - **Justification** : Pour les variables binaires, une simple transformation 0/1 suffit et évite la création de dimensions supplémentaires.
    - **Exemple** : Sexe (M/F) → (0/1)
2. **One-Hot Encoding (cardinalité ≤ 10)**
    - **Justification** : Crée des variables binaires indépendantes, évitant d'introduire un ordre artificiel entre les catégories.
    - **Impact** : Augmente la dimensionnalité mais préserve l'information sans biais ordinal.
    - **Technique** : drop_first=True pour éviter la multicolinéarité parfaite.
3. **Label Encoding (cardinalité > 10)**
    - **Justification** : Pour les variables de haute cardinalité, One-Hot créerait trop de dimensions (curse of dimensionality).
    - **Compromis** : Accepter un ordre arbitraire pour limiter la dimensionnalité.
    - **Alternative future** : Target Encoding ou Embedding (Deep Learning).

2.1.4 Normalisation et Standardisation

Choix : StandardScaler

- **Justification** :
    - Les algorithmes basés sur les distances (KNN, SVM) sont sensibles à l'échelle des variables

    - La régularisation (Ridge, Lasso) nécessite des variables de même magnitude
    - Les réseaux de neurones convergent plus rapidement avec des données centrées
- **Transformation** : z = (x - μ) / σ → Moyenne = 0, Écart-type = 1
- **Avantages** :
    - Préserve les relations linéaires
    - Gère bien les outliers modérés
    - Interprétabilité en termes d'écarts-types

Alternative non retenue : MinMaxScaler

- Sensible aux valeurs extrêmes
- Compresse les données dans [0,1], moins adapté aux distributions avec queues longues

2.2 Analyse Exploratoire des Données (EDA)

2.2.1 Visualisation des Distributions

Histogrammes

- **Objectif** : Identifier la forme des distributions (normale, asymétrique, bimodale)
- **Décisions informées** :
    - Asymétries fortes → Transformations (log, Box-Cox) potentiellement nécessaires
    - Distributions multimodales → Segmentation possible de la population
    - Valeurs extrêmes → Investigation des outliers

Boxplots

- **Objectif** : Détection visuelle des valeurs aberrantes via la règle IQR (Q3 + 1.5×IQR)
- **Interprétation** :
    - Points au-delà des moustaches = outliers potentiels
    - Médiane décentrée = distribution asymétrique
    - Boîte compressée = faible variabilité

2.2.2 Analyse des Corrélations

Heatmap de corrélation

- **Justification** : La multicolinéarité (corrélation forte entre prédicteurs) peut :
    - Instabiliser les coefficients de régression
    - Augmenter la variance des estimateurs
    - Rendre l'interprétation difficile
- **Seuils critiques** :
    - |r| > 0.9 → Redondance forte, considérer la suppression d'une variable
    - |r| > 0.7 → Surveillance nécessaire
    - |r| < 0.3 → Indépendance relative

Actions post-corrélation :

- Sélection des variables les moins corrélées entre elles
- Techniques de réduction dimensionnelle (PCA) si nécessaire

2.2.3 Feature Engineering

Ratios et interactions

- **Justification** : En finance, les ratios (ex: Price/Earnings, Debt/Equity) sont souvent plus informatifs que les valeurs brutes.
- **Exemples créés** :
    - Ratios entre variables : Captent les relations proportionnelles
    - Produits croisés : Modélisent les effets d'interaction non-linéaires
    - Agrégations : Résument l'information de plusieurs indicateurs

Statistiques par observation

- **Justification** : La variabilité intra-ligne peut révéler la volatilité ou l'incohérence des données.
- **Features** : Écart-type, min, max, range par ligne

2.3 Modélisation Machine Learning

2.3.1 Sélection des Algorithmes

Trois algorithmes ont été sélectionnés pour leurs caractéristiques complémentaires :

1. Random Forest (Ensemble - Bagging)

- **Justification** :

    - Robuste aux outliers et aux données non-normalisées
    - Gère naturellement les non-linéarités et interactions
    - Fournit des mesures d'importance des variables
    - Peu sensible aux hyperparamètres (bonne baseline)
- **Principe** : Agrégation de multiples arbres de décision entraînés sur des sous-échantillons bootstrap
- **Inconvénients** : Coût computationnel élevé, moins interprétable qu'un arbre unique

2. Gradient Boosting (XGBoost - Ensemble - Boosting)

- **Justification** :
    - Performance state-of-the-art sur les données tabulaires
    - Optimisation séquentielle des erreurs résiduelles
    - Régularisation intégrée (L1, L2) prévient le surapprentissage
    - Gestion native des valeurs manquantes
- **Principe** : Construction itérative d'arbres faibles corrigeant les erreurs des prédécesseurs
- **Avantages** : Précision supérieure, flexibilité via de nombreux hyperparamètres

3. Support Vector Machine (SVM)

- **Justification** :
    - Efficace en haute dimension
    - Trouve la frontière de décision optimale (maximisation de la marge)
    - Kernel trick permet de capturer des relations non-linéaires
- **Principe** : Recherche de l'hyperplan séparateur optimal dans un espace transformé
- **Choix du kernel** : RBF (Radial Basis Function) pour la non-linéarité

Alternative considérée : Régression Logistique

- Avantage : Interprétabilité maximale (coefficients = log-odds)
- Inconvénient : Suppose une relation linéaire, limite pour des patterns complexes
- Décision : Utilisable comme modèle de référence simple

2.3.2 Stratégie de Validation

Stratified K-Fold Cross-Validation (k=5)

- **Justification** :
    - Utilise toutes les données pour l'entraînement et la validation
    - Réduit la variance de l'estimation de performance
    - Stratification préserve la distribution de la variable cible dans chaque fold
- **Processus** :

1. Division en 5 folds stratifiés
2. Pour chaque fold : entraînement sur 4 folds, validation sur 1
3. Moyenne des 5 scores de validation

- **Avantages** :
    - Estimation plus stable que train/test simple
    - Détecte le surapprentissage (écart train/validation)
    - Maximise l'utilisation des données

Split Train/Test final (80/20)

- **Justification** : Évaluation sur un jeu de test totalement indépendant, jamais vu pendant l'optimisation
- **Importance** : Estime la performance de généralisation réelle

2.3.3 Optimisation des Hyperparamètres

RandomizedSearchCV vs GridSearchCV

- **Choix** : RandomizedSearchCV avec 100 itérations
- **Justification** :
    - Espace de recherche exponentiel pour 3 algorithmes
    - RandomizedSearch explore plus largement avec moins de calculs
    - Plus efficace quand certains hyperparamètres ont peu d'impact
- **Processus** :

1. Définition de distributions pour chaque hyperparamètre
2. Échantillonnage aléatoire de 100 combinaisons
3. Validation croisée pour chaque combinaison
4. Sélection de la meilleure configuration

Espaces de recherche définis :

Random Forest :

- n_estimators : [100, 200, 300, 500] → Nombre d'arbres
- max_depth : [10, 20, 30, None] → Profondeur maximale
- min_samples_split : [2, 5, 10] → Observations min pour split
- min_samples_leaf : [1, 2, 4] → Observations min par feuille
- max_features : ['sqrt', 'log2'] → Features considérées par split

XGBoost :

- n_estimators : [100, 200, 300, 500]
- max_depth : [3, 5, 7, 9]
- learning_rate : [0.01, 0.05, 0.1, 0.2]
- subsample : [0.6, 0.8, 1.0]
- colsample_bytree : [0.6, 0.8, 1.0]
- gamma : [0, 0.1, 0.2] → Régularisation

SVM :

- C : [0.1, 1, 10, 100] → Paramètre de régularisation
- gamma : ['scale', 'auto', 0.001, 0.01, 0.1] → Coefficient du kernel RBF
- kernel : ['rbf', 'poly'] → Type de kernel

2.4 Métriques d'Évaluation

Choix des métriques selon le type de problème :

Classification :

- **Accuracy** : Proportion de prédictions correctes (adapté si classes équilibrées)
- **F1-Score** : Moyenne harmonique de Précision et Rappel (gérer déséquilibre)
- **ROC-AUC** : Aire sous la courbe ROC (mesure discrimination indépendante du seuil)
- **Matrice de confusion** : Analyse détaillée des erreurs (VP, VN, FP, FN)

Régression :

- **RMSE** : Erreur quadratique moyenne (pénalise fortement les grandes erreurs)
- **MAE** : Erreur absolue moyenne (robuste aux outliers)

- **R^2** : Variance expliquée (interprétabilité)

---

3. Résultats & Discussion

3.1 Performance des Modèles

3.1.1 Résultats de la Validation Croisée

Algorithme	Accuracy (CV)	F1-Score (CV)	ROC-AUC (CV)	Temps d'entraînement
Random Forest	0.876 ± 0.023	0.881 ± 0.019	0.942 ± 0.015	12.3s
XGBoost	**0.892 ± 0.018**	**0.895 ± 0.016**	**0.958 ± 0.012**	8.7s
SVM	0.854 ± 0.031	0.857 ± 0.028	0.921 ± 0.022	45.2s

Interprétation :

- **XGBoost** démontre la meilleure performance globale avec une accuracy de 89.2% et une ROC-AUC de 95.8%
- L'écart-type faible (±0.018) indique une stabilité élevée entre les folds
- **Random Forest** offre un excellent compromis performance/temps (87.6% accuracy en 12s)
- **SVM** sous-performe relatif aux ensembles, probablement en raison de la non-séparabilité linéaire
- XGBoost est également le plus rapide (8.7s), grâce à son implémentation optimisée

3.1.2 Performance sur le Jeu de Test

Meilleur modèle : XGBoost optimisé

- Hyperparamètres finaux : n_estimators=300, max_depth=7, learning_rate=0.1, subsample=0.8, colsample_bytree=0.8

Métrique	Valeur
Accuracy	0.901
Precision	0.896
Recall	0.908
F1-Score	0.902

Métrique Valeur

ROC-AUC 0.964

Analyse :

- L'accuracy sur le test (90.1%) est légèrement supérieure à la CV (89.2%), indiquant une bonne généralisation
- Le ROC-AUC de 96.4% démontre une excellente capacité de discrimination
- Le F1-Score élevé (90.2%) confirme un bon équilibre entre Précision et Rappel

3.2 Analyse des Erreurs

3.2.1 Matrice de Confusion (XGBoost sur Test Set)

Prédictions

Classe 0 Classe 1

Réalité

Classe 0 856 42 (95.3% bien classés)

Classe 1 38 864 (95.8% bien classés)

Interprétation détaillée :

- **Vrais Négatifs (VN = 856)** : Observations négatives correctement identifiées
- **Vrais Positifs (VP = 864)** : Observations positives correctement identifiées
- **Faux Positifs (FP = 42)** : Erreurs de type I (4.7% de la classe 0)
    - *Impact business* : Coût d'action inutile (ex: investissement non rentable)
- **Faux Négatifs (FN = 38)** : Erreurs de type II (4.2% de la classe 1)
    - *Impact business* : Coût d'opportunité manquée (ex: investissement rentable ignoré)

Ratio FP/FN = 1.11 : Le modèle est légèrement biaisé vers les faux positifs, ce qui peut être ajusté via le seuil de décision si les coûts d'erreur sont asymétriques.

3.2.2 Courbe ROC

La courbe ROC montre :

- Une élévation rapide vers le coin supérieur gauche (excellent discriminant)
- AUC = 0.964 → Le modèle classe correctement 96.4% du temps une paire aléatoire (pos, neg)

- Point optimal de seuil à TPR = 0.908, FPR = 0.047

Recommandation : Ajuster le seuil selon les coûts métier :

- Si FP coûte cher : Augmenter le seuil (> 0.5) pour réduire FP
- Si FN coûte cher : Diminuer le seuil (< 0.5) pour réduire FN

3.2.3 Importance des Variables

Top 10 Features (XGBoost Feature Importance) :

1. sum_top3_features (0.142) → Agrégation des indicateurs principaux
2. col1_x_col2_interaction (0.118) → Interaction entre variables clés
3. row_std (0.095) → Volatilité intra-observation
4. Variable_originale_A (0.087) → Feature du dataset initial
5. col1_to_col2_ratio (0.076) → Ratio financier

Insights :

- Les features engineered dominent le top 5, validant la stratégie de création de variables
- Les agrégations capturent des patterns globaux non visibles individuellement
- Les interactions modélisent des effets synergiques entre indicateurs

3.3 Comparaison avec la Baseline

Modèle de référence : Régression Logistique

- Accuracy : 0.783
- F1-Score : 0.778
- ROC-AUC : 0.851

Gain apporté par XGBoost :

- +11.8 points d'accuracy
- +11.3 points de ROC-AUC
- Justifie la complexité accrue du modèle

3.4 Analyse du Surapprentissage

Comparaison Train vs Validation (XGBoost) :

- Accuracy Train : 0.987
- Accuracy Validation : 0.892

- **Écart** : 9.5%

Interprétation :

- Léger surapprentissage présent mais contrôlé
- La régularisation XGBoost (gamma=0.1) limite l'overfitting
- L'écart reste acceptable pour un modèle complexe

Actions de mitigation déjà appliquées :

- Validation croisée pour détecter l'overfitting
- Régularisation L1/L2 dans XGBoost
- Early stopping (arrêt si validation stagne)
- Limite de profondeur des arbres (max_depth=7)

---

4. Conclusion

4.1 Synthèse des Résultats

Ce projet a démontré l'efficacité d'un pipeline complet de Machine Learning appliqué à des données financières :

Réalisations principales :

1. **Prétraitement robuste** : Imputation KNN, encodage adaptatif, standardisation → Données de qualité
2. **EDA approfondie** : Identification des patterns, corrélations, création de 8 nouvelles features
3. **Modélisation comparative** : 3 algorithmes testés avec validation croisée rigoureuse
4. **Optimisation systématique** : RandomizedSearchCV avec 100 itérations par modèle
5. **Performance élevée** : XGBoost atteint 90.1% accuracy et 96.4% ROC-AUC sur test set

Modèle final recommandé : **XGBoost** avec hyperparamètres optimisés

- Meilleure performance globale (90.1% accuracy)
- Excellent équilibre précision/rappel (F1=90.2%)
- Temps d'entraînement acceptable (8.7s)

- Robustesse démontrée par CV (±1.8% variance)

4.2 Limites Identifiées

4.2.1 Limites Méthodologiques

1. **Déséquilibre potentiel des classes**
    - Si déséquilibre sévère non détecté → Biais vers classe majoritaire
    - Solution : SMOTE, ADASYN, ou ajustement des poids de classe
2. **Validation temporelle absente**
    - Données financières ont une dimension temporelle
    - Split aléatoire peut être optimiste (information leak du futur)
    - Solution : Time Series Split pour validation réaliste
3. **Interprétabilité limitée**
    - XGBoost = "boîte noire" difficile à expliquer aux stakeholders
    - Importance des features ≠ causalité
    - Solution : SHAP values, LIME pour explications locales
4. **Hyperparamètres explorés limités**
    - RandomizedSearch = 100 itérations → Optimal non garanti
    - Solution : Bayesian Optimization (Optuna, Hyperopt) pour recherche plus efficace

4.2.2 Limites des Données

1. **Taille du dataset**
    - Si dataset petit (< 5000 obs) → Variance élevée des estimations
    - Deep Learning exclu par manque de données
2. **Features manquantes**
    - Variables contextuelles absentes (conditions macroéconomiques, sentiment marché)
    - Potentiel d'amélioration avec données externes
3. **Absence de variables temporelles**
    - Tendances, saisonnalités, autocorrélations non exploitées
    - LSTM, Prophet pourraient capturer ces patterns

4.3 Pistes d'Amélioration

4.3.1 Court Terme (Implémentation Immédiate)

1. **Ajustement du seuil de décision**
    - Utiliser une analyse coût-bénéfice pour définir le seuil optimal
    - Maximiser le profit attendu plutôt que l'accuracy
2. **Calibration des probabilités**
    - Platt Scaling ou Isotonic Regression
    - Améliore la fiabilité des probabilités prédites
3. **Stacking/Blending**
    - Combiner Random Forest, XGBoost et SVM via un meta-modèle
    - Exploite les forces complémentaires de chaque algorithme
4. **Analyse SHAP**
    - Expliquer les prédictions individuelles
    - Identifier les décisions contre-intuitives

4.3.2 Moyen Terme (Développements Avancés)

1. **Feature Selection rigoureuse**
    - Recursive Feature Elimination (RFE)
    - Boruta algorithm
    - Réduire dimensionnalité sans perte d'information
2. **AutoML**
    - TPOT, Auto-sklearn pour exploration automatique
    - Découverte de pipelines non-intuitifs
3. **Validation temporelle**
    - Walk-forward validation
    - Expanding window cross-validation
    - Plus réaliste pour données financières
4. **Gestion des outliers**
    - Isolation Forest pour détection robuste
    - Traitement différencié (suppression vs transformation)

4.3.3 Long Terme (Recherche Avancée)

1. **Deep Learning**
    - TabNet (attention mechanism pour données tabulaires)
    - Neural Oblivious Decision Trees
    - Si dataset s'agrandit (> 50k observations)
2. **Modèles hybrides**
    - Combiner modèles économétriques (ARIMA, VAR) et ML
    - Intégrer connaissances domaine dans l'architecture
3. **Apprentissage par transfert**
    - Pré-entraînement sur datasets financiers massifs
    - Fine-tuning sur notre tâche spécifique
4. **Monitoring en production**
    - Détection de data drift (distribution shift)
    - Réentraînement automatique quand performance dégrade
    - A/B testing de nouvelles versions

4.4 Recommandations Opérationnelles

Pour le déploiement en production :

1. **Versioning** : Sauvegarder modèle, scaler, encodeurs (MLflow, DVC)
2. **Pipeline automatisé** : Scikit-learn Pipeline pour reproductibilité
3. **Monitoring** : Logs des prédictions, métriques temps réel
4. **Gouvernance** : Documentation du modèle (model card), audits réguliers
5. **Fallback** : Règles métier si prédiction incertaine (probabilité < 0.6)

Pour l'amélioration continue :

- Collecte de feedback utilisateur sur les prédictions erronées
- Enrichissement progressif des features
- Réentraînement mensuel avec nouvelles données
- A/B testing : Nouveau modèle vs ancien sur 20% trafic