

Interprétation des Résultats

1. Évaluation des Performances du Modèle

Après avoir utilisé **GridSearchCV** pour optimiser les paramètres du modèle Random Forest, J'ai trouvé les meilleurs paramètres suivants :

- **Profondeur des arbres (max_depth)** : Aucun limite, ce qui permet aux arbres de se développer autant que nécessaire.
- **Nombre d'arbres (n_estimators)** : 100 arbres.
- **Critère pour diviser un nœud (min_samples_split)** : 2 échantillons minimum.
- **Nombre minimum d'échantillons dans une feuille (min_samples_leaf)** : 1 échantillon.
- **Caractéristiques à prendre en compte (max_features)** : la racine carrée du nombre total de caractéristiques.

Le meilleur score obtenu pendant la GridSearch est de **98.7%** sur la validation croisée, ce qui montre que le modèle est très performant.

2. Score d'Accuracy

L'accuracy sur les données de test est de **94.8%**, ce qui signifie que le modèle a fait la prédiction correcte dans environ 95% des cas.

3. Rapport de Classification

- **Classe 0 (négative) :**
 - **Précision** : 97% — Cela veut dire que quand le modèle prédit la classe 0, il est correct 97% du temps.
 - **Rappel** : 91% — Cela signifie que parmi toutes les instances réelles de la classe 0, 91% ont été correctement identifiées par le modèle.
 - **F1-score** : 94% — Une mesure qui combine précision et rappel.
- **Classe 1 (positive) :**

- **Précision** : 94% — Quand le modèle prédit la classe 1, il est correct 94% du temps.
- **Rappel** : 98% — Parmi toutes les instances réelles de la classe 1, 98% ont été correctement identifiées.
- **F1-score** : 96% — Très bon équilibre entre précision et rappel.

Le **score moyen** (calculé en prenant la moyenne des résultats pour chaque classe) est de **95%**, ce qui montre que le modèle fonctionne bien dans l'ensemble.

4. Analyse des Erreurs

- Le modèle est particulièrement **bon pour identifier les instances de la classe 1**, avec un très bon **rappel de 98%**. Par contre, il est un peu moins performant pour la classe 0, avec un rappel de **91%**, ce qui veut dire qu'il manque parfois des instances négatives.
- En termes de **précision**, la classe 0 est légèrement meilleure que la classe 1, car plus de prédictions pour la classe 0 sont correctes.

5. Améliorations Proposées

- Pour améliorer la performance du modèle, j'ai essayé d'ajuster le **seuil de classification** et utiliser des techniques comme le **suréchantillonnage** ou le **sous-échantillonnage** pour mieux traiter le déséquilibre entre les classes.

6. Comparaison des Résultats

J'ai comparé les résultats du modèle après optimisation des hyperparamètres avec ceux du modèle initial, je peux voir que l'optimisation a amélioré les performances. Le score de validation croisée a augmenté à **98.7%**, et l'accuracy sur les données de test a atteint **94.8%**. Les scores F1 pour les deux classes sont également bons, ce qui montre que le modèle est équilibré entre précision et rappel.

Conclusion

Donc le modèle est très performant, avec une **précision de 94.8%** et un bon **équilibre entre les classes**. Il fonctionne très bien. (Excellent)

```
Best parameters: {'rf_max_depth': None, 'rf_max_features': 'sqrt', 'rf_min_samples_leaf': 1, 'rf_min_samples_split': 2, 'rf_n_estimators': 100}
Best cross-validation score: 0.9869918699186991
Accuracy: 0.948851948851948
Classification Report:
precision    recall    f1-score   support
          0       0.97     0.91      0.94      64
          1       0.94     0.98      0.96      90

   accuracy         0.95      0.94      0.95     154
  weighted avg     0.95      0.95      0.95     154
```

Voici Aussi La Matrice De Corrélation Avant De Lancer L'apprentissage ! :=)

