Projet de mise en œuvre d'un modèle de Machine Learning avec Scikit-learn - Titanic Dataset

Le dataset Titanic est un classique du Machine Learning, utilisé pour prédire si un passager a survécu ou non au naufrage du Titanic en 1912. L'objectif est d'utiliser les données disponibles pour entraîner un modèle qui peut faire cette prédiction. Le fichier contient 891 lignes (passagers) et 12 colonnes représentant différentes caractéristiques, telles que le sexe, l'âge, la classe du billet, le tarif payé, et le port d'embarquement.

Description du dataset :

- **Nombre total d'entrées :** 891
- **Colonnes :**

o PassengerId : Identifiant unique des passagers.

o Survived : Indique si le passager a survécu (1) ou non (0).

o Pclass : Classe du billet (1ère, 2ème ou 3ème classe).

o Name : Nom du passager.

o Sex : Sexe du passager.

o Age : Âge du passager (avec des valeurs manquantes).

o SibSp : Nombre de frères, sœurs ou conjoint à bord.

o Parch : Nombre de parents ou enfants à bord.

o Ticket : Numéro de billet.

o Fare : Tarif payé.

o Cabin : Numéro de cabine (fortement manquant).

o Embarked : Port d'embarquement (C = Cherbourg, Q = Queenstown, S = Southampton).

Préparation des données :

1. Valeurs manquantes :

La colonne Age contient des valeurs manquantes pour 177 passagers (~20%). Les valeurs manquantes sont remplacées par la médiane des âges.

La colonne Embarked manque pour 2 passagers. Remplissage des ports d'embarquement manquants par la mode

La colonne Cabin est fortement incomplète (seulement 204 valeurs non nulles) et a été supprimée pour simplifier l'analyse.

2. Renommage des colonnes :

Les colonnes ont été renommées pour être plus lisibles et plus significatives, comme :

- PassengerId: ID,
- Survived: Survie,
- Pclass: Classe,
- Name: Nom,
- Sex: Sexe,
- SibSp: Nb_Frères_Soeurs,
- Parch: Nb_Parents_Enfants,
- Ticket: Billet,
- Fare: Tarif,
- Embarked: Embarquement

3. Encodage et normalisation :

- Les variables catégoriques (Sexe, Embarked) ont été transformées en valeurs numériques.
- Les colonnes numériques comme Âge et Tarif ont été normalisées pour faciliter l'entraînement des modèles.

4. Modèles de Machine Learning

Les données ont été divisées (80 : 20) en entraînement et test.

Quatre algorithmes ont été testés :

a) **Random Forest** : Un modèle d'ensemble basé sur plusieurs arbres de décision.

b) **Decision Tree** : Simple et interprétable, mais souvent moins précis.
c) **SVM (Support Vector Machine)** : Modèle puissant pour des données bien séparables.
d) **Régression Logistique** : Classique pour les tâches de classification binaire.

Chaque modèle a été évalué sur un jeu de test en utilisant des métriques comme la précision, le rappel, le score F1.

Résultats obtenus :

Les performances des modèles sont résumées ci-dessous :

Modèle	Précision	Précision Moy.	Rappel Moy.	F1-Score Moy.
Random Forest	82%	81%	82%	81%
Decision Tree	78%	77%	78%	78%
SVM	82%	81%	82%	81%
Régression Logistique	81%	80%	81%	81%

Le **Random Forest** a offert les meilleures performances globales. ***Précision*** : C'est la proportion de prédictions correctes parmi toutes les prédictions positives faites. ***Rappel (Sensibilité)*** : C'est la proportion de vrais positifs parmi tous les cas réels positifs.

Visualisation des résultats :

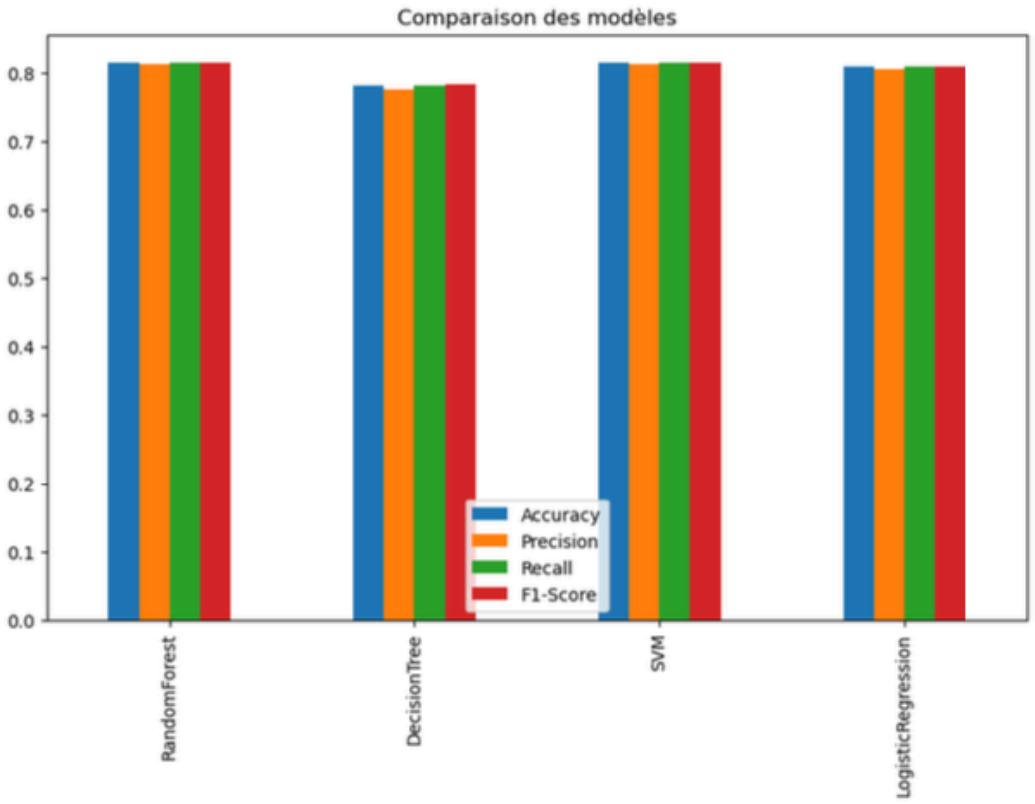

Figure 1 : Figure comparatif des résultats obtenues de l'entrainement de quatre modéles de machine Learning

Résultat détaillé pour le Random Forest :

Tableau 1 : Tableau représentant les résultats obtenus pour des prédictions du modèle Random forest

Classe	Précision	Rappel	F1-Score	Support
0	0.83	0.87	0.85	105
1	0.80	0.74	0.77	74
Total (Accuracy)	-	-	**0.82**	**179**
Macro Moy.	0.81	0.80	0.81	179
Weighted Moy.	0.81	0.82	0.81	179

Matrice de confusion :

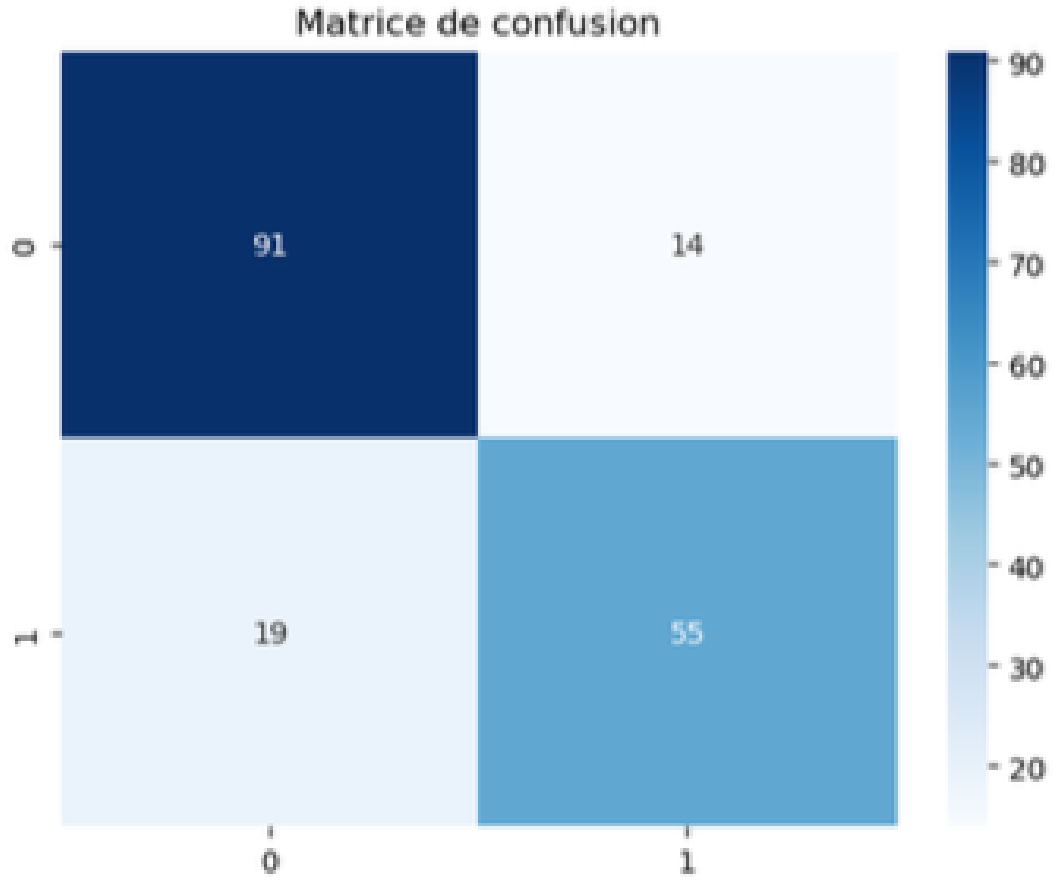

Figure 2 : Matrice de confusion pour les détections du modèle Random Forest

Interprétation des éléments :

a) **Vrais négatifs (91)** : Ce sont les cas où le modèle a correctement prédit que la personne n'a pas survécu.

b) **Faux positifs (14)** : Ce sont les cas où le modèle a prédit que la personne avait survécu, alors qu'en réalité elle n'a pas survécu.

c) **Faux négatifs (19)** : Ce sont les cas où le modèle a prédit que la personne n'avait pas survécu, alors qu'en réalité elle avait survécu.

d) **Vrais positifs (55)** : Ce sont les cas où le modèle a correctement prédit que la personne avait survécu.

Le modèle montre de bonnes performances globales, mais il y a encore des faux négatifs (19), ce qui suggère qu'il pourrait y avoir un risque de ne pas prédire correctement certaines survivances. Les faux positifs (14) sont également relativement faibles, ce qui signifie que le modèle n'attribue pas trop de survies erronées.

Courbe ROC :

La courbe ROC du modèle Random Forest montre une AUC (Area Under Curve) de 0.90, indiquant une excellente capacité de séparation entre les classes.

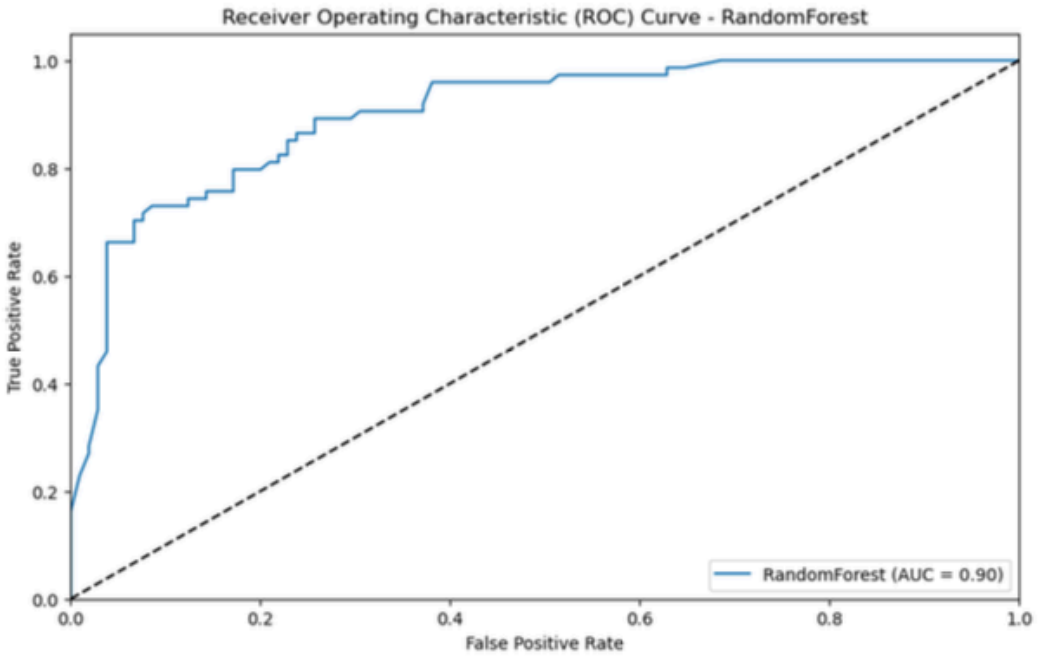

Figure 3 : Courbe ROC modèle Random Forest

Optimisation des hyperparamètres :

Une recherche par grille (GridSearchCV) a été effectuée pour optimiser les hyperparamètres du modèle Random Forest. Les meilleurs paramètres trouvés sont :

- Nombre d'arbres (n_estimators) : 200
- Critère de division (criterion) : Entropy
- Minimum d'échantillons pour diviser un nœud (min_samples_split) : 5

Tableau 2 : Résultats du modèle Random forest après réglage des hyperparamètres

Classe	Précision	Rappel	F1-Score	Support
0	0.83	0.90	0.86	105
1	0.83	0.74	0.79	74
Total (Accuracy)	**-**	**-**	**0.83**	**179**
Macro Moy.	0.83	0.82	0.82	179
Weighted Moy.	0.83	0.83	0.83	179

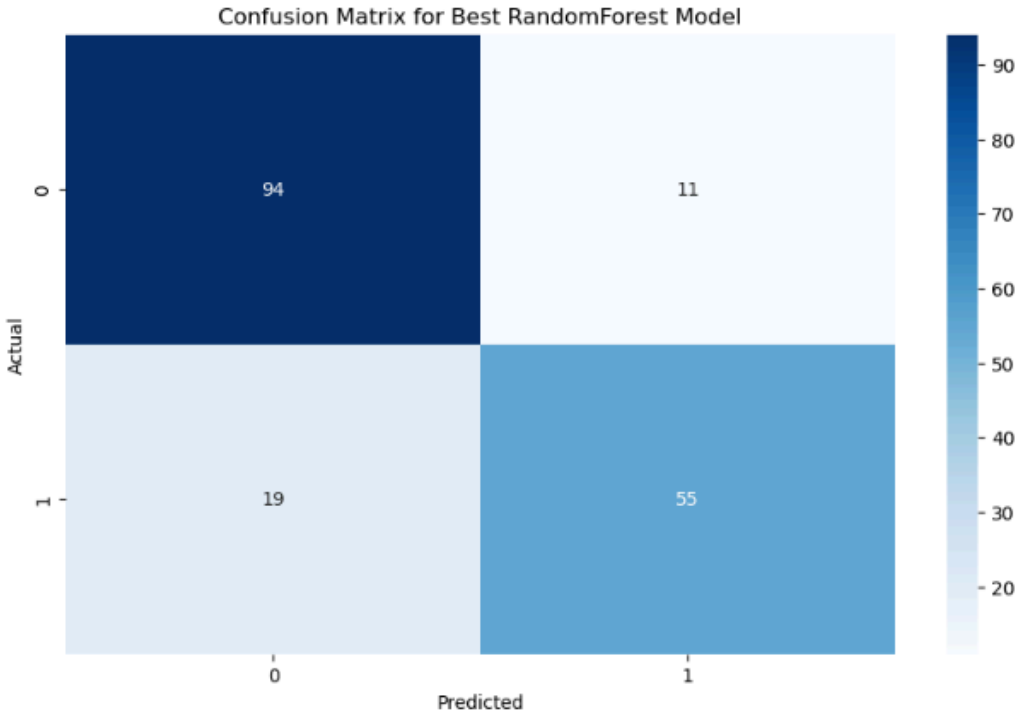

Figure 4 : Matrice de confusion du modèle Random Forest après choix des hyperparamètres

Après un bon choix des hyperparamètres, on remarque une amélioration de performance de prédiction.

Analyse globale :

Avantages :

1. **Approche complète de prétraitement des données:**

La gestion des valeurs manquantes (remplissage de l'âge par la médiane et de l'embarquement par la mode, suppression de la colonne Cabin) est appropriée pour éviter les biais dans l'entraînement du modèle. Ces choix sont communs dans ce genre de dataset où une grande partie des données peut manquer.

2. **Modèles performants:**

Le modèle Random Forest a montré une performance solide, avec un **score AUC** élevé (0.90), ce qui signifie qu'il est capable de bien distinguer les classes "Survécu" et "Non-survécu". Il a également surpassé les autres modèles en termes de précision et de rappel, ce qui le rend robuste pour cette tâche de classification.

3. **Choix d'algorithmes variés:**

L'évaluation de plusieurs modèles de Machine Learning permet de comparer différentes approches et de choisir le meilleur. Cette diversité d'approches garantit que le modèle final est bien choisi après un processus d'évaluation minutieux.

Limites :

1. **Manque de données représentatives :**
Le dataset de Titanic contient seulement 891 observations. Bien que ce nombre soit suffisant pour un problème de classification simple, un plus grand volume de données pourrait améliorer la généralisation du modèle et minimiser les risques de surapprentissage (overfitting).

2. **Problème de biais sur certaines variables :**

La colonne Sexe peut contenir des biais. Par exemple, la survie était historiquement plus élevée chez les femmes que chez les hommes, ce qui pourrait fausser la prédiction si les modèles ne sont pas correctement ajustés pour compenser ce biais. De même, la classe sociale (Pclass) peut jouer un rôle déterminant dans la survie, ce qui pourrait introduire des biais socio- économiques dans les prédictions si ce facteur est mal équilibré dans le dataset.

3. **Sous-représentation des classes:**
La classe des survivants (Survived = 1) est souvent beaucoup plus petite que la classe des non-survivants. Cela peut entraîner un déséquilibre dans les prédictions. Bien que l'évaluation via AUC soit robuste aux déséquilibres de classe, une meilleure gestion de ces déséquilibres pourrait améliorer les performances des modèles, notamment pour la classe minoritaire.

Pistes d'améliorations :

1. **Exploration de modèles plus complexes** :

Tester des modèles de **boosting** (XGBoost, LightGBM, CatBoost) pour des performances supérieures.

2. **Amélioration du traitement des données manquantes** :

Appliquer des techniques de remplissage plus avancées (régression, KNN) pour les valeurs manquantes, notamment pour l'âge et l'embarquement.

3. **Augmentation des données et traitement des classes déséquilibrées :**

- Utiliser des techniques comme pour équilibrer les classes minoritaires et majoritaires.
- Ajouter des **caractéristiques supplémentaires** (socio-économiques, interactions entre passagers, etc.) pour enrichir le modèle.