Rapport détaillé

Participation à une compétition Kaggle

Spaceship Titanic 

Auteur : Arslane Lahmer

Sommaire

I. Introduction

Pour le projet final du parcours Machine Learning, nous prenons part à la compétition Kaggle ***Spaceship Titanic***.

Il s'agit d'un problème de classification binaire. Le but est de déterminer si un passager a été transporté ou non, dans une dimension parallèle à la suite d'une collision du vaisseau dans une faille spatio-temporelle. Pour ce faire, nous disposons d'un jeu de données récupéré sur l'ordinateur de bord du vaisseau.

Le jeu de données nous fournit les informations suivantes :

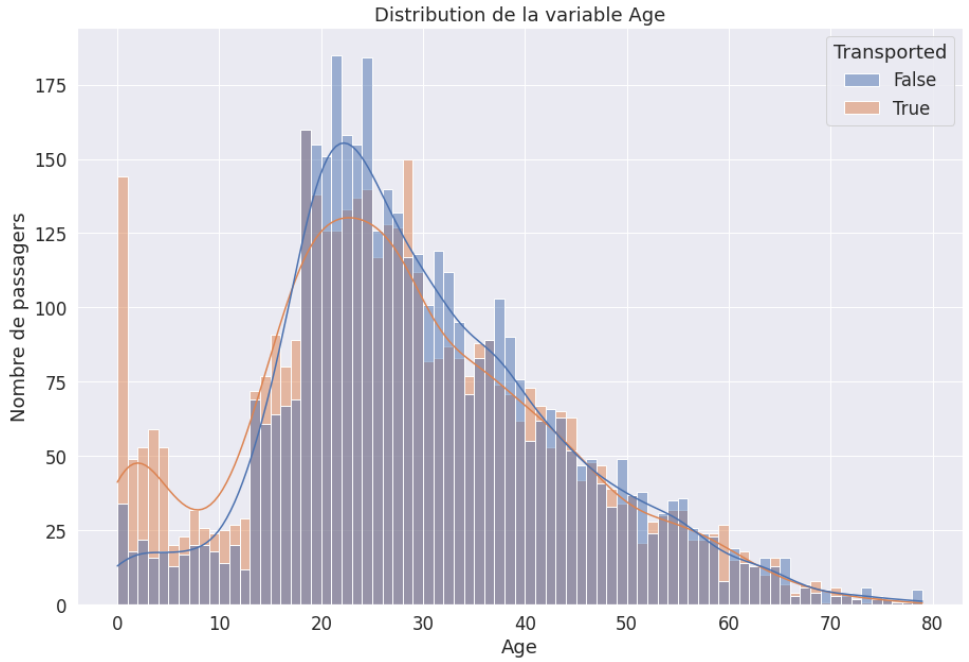
- **PassengerId** - Un identifiant unique pour chaque passager. prend la forme gggg_pp où gggg correspond au groupe avec lequel le passager voyage tandis que pp représente leur nombre au sein du groupe.
- **HomePlanet** - La planète d'où a lieu le départ du passager.
- **CryoSleep** - Indique si le passager a choisi d'être plongé dans un sommeil cryogénique pendant le voyage.
- **Cabin** - Le numéro de cabine du passager.
- **Destination** - La planète où le passager va débarquer.
- **Age** - Âge du passager.
- **VIP** - Si le passager a payé pour des services VIP.
- **RoomService**, **FoodCourt**, **ShoppingMall**, **Spa**, **VRDeck** - Montant dépensé par le passager pour chaque service mis à disposition sur le *Spaceship Titanic*'s.
- **Name** - Nom et prénom du passager.
- **Transported** - Indique si le passager a été transporté dans une dimension parallèle. Ce qui représente la variable cible .

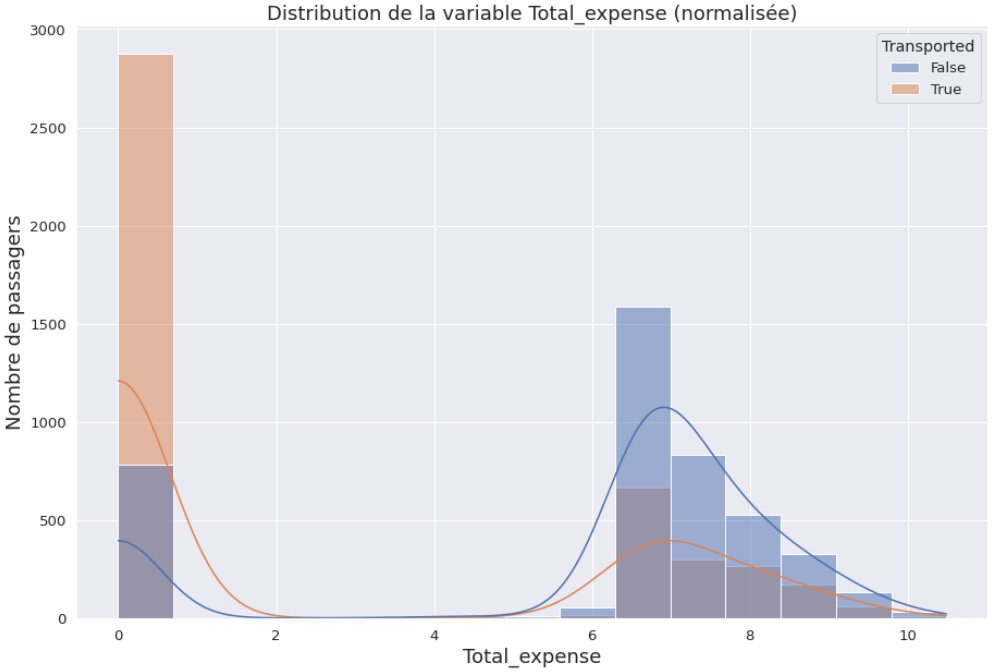
Le jeu d'entraînement compte 8693 entrées tandis que le jeu de test en compte 4277.

II. Analyse exploratoire

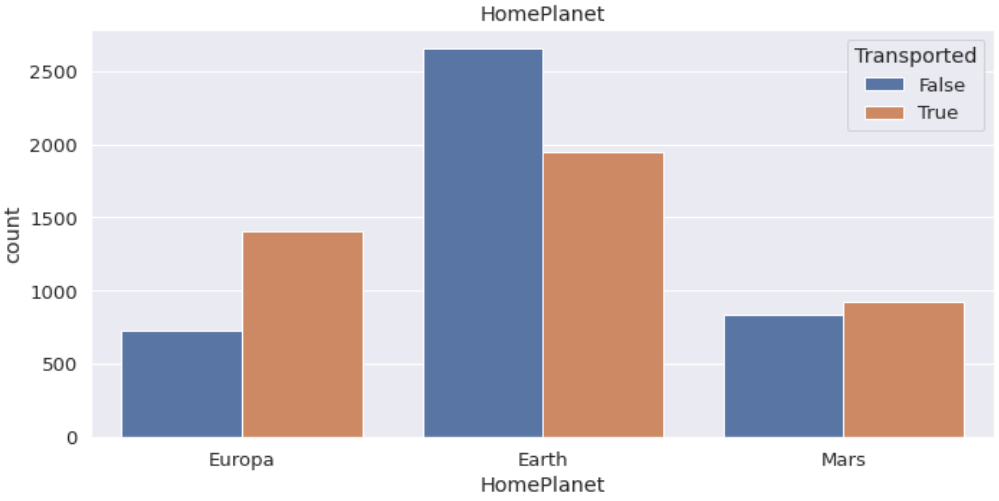
L'analyse exploratoire nous permettra d'explorer et d'analyser le jeu de données en décrivant le comportement des variables, leurs importance respective quand il s'agit de prédire/classer. Cela nous aidera dans le processus de préparation des données. On va visualiser les graphes de quelques unes des variables les plus importantes quand il s'agit de prédire si le passager a été transporté ou non.

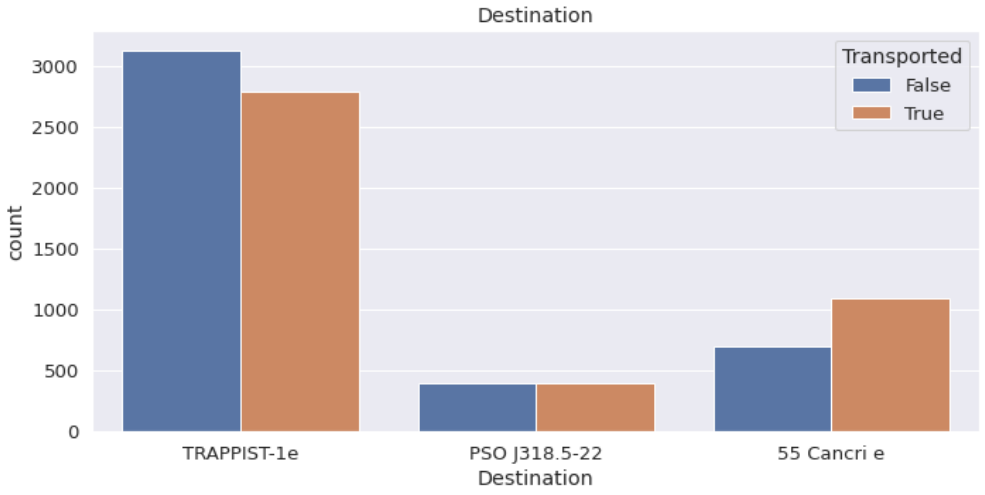
A. Variables numériques

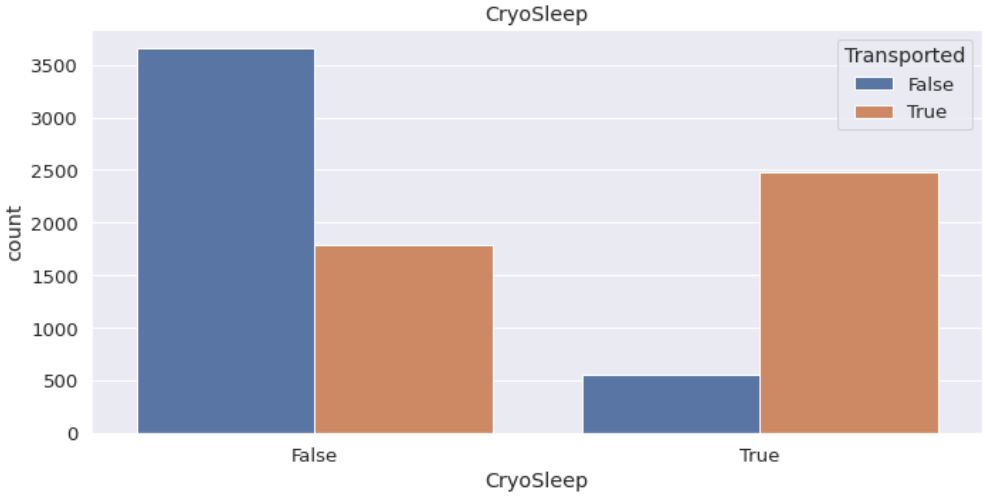




B. Variables catégorielles







V. Feature engineering

Consiste à créer des variables à partir du jeu de données existant afin d'améliorer l'apprentissage du modèle.

Variables créées :

- **Total_expense** - Représente le montant total des dépenses de chaque passager
- **Spent_money** - Indique si le passager a effectué des dépenses ou pas.
- **Deck** - Pont de la cabine
- **Side** - Côté de la cabine

VI. Prétraitement des donnés

On impute d'abord les données manquantes grâce aux forêts aléatoires. Une fois les features qu'on juge utiles pour la prédiction sélectionnées, on procède à ce qui suit :

- La normalisation des variables numériques à l'aide du log(n+1) afin d'avoir des données plus homogènes.
- L'encodage des variables catégorielles à l'aide de CatBoostEncoder.

Features sélectionnées : "CryoSleep", "Spent_money", "Age","RoomService","FoodCourt","ShoppingMall","Spa", "VRDeck", "Total_expense", "HomePlanet", "Destination", "Deck", "Side".

Le jeu de données est désormais prêt à l'emploi.

VII. Implémentation du modèle TPOT

Nous allons utiliser TPOT pour implémenter le modèle qui servira à la prédiction.

TPOT est un outil d'Auto ML qui se base sur les arbres de décision et qui a recours à la programmation génétique pour trouver le modèle de pipelines le plus performant.

Une fois le jeu de données prêt à l'emploi, TPOT aide à :

- Sélection des features
- Sélection meilleur modèle
- Optimiser ses hyperparamètres

VIII. Performances du modèle

Le modèle sélectionné est un RandomForestClassifier optimisé. En validation croisée sur le jeu d'entraînement on obtient les résultats suivants :

Accuracy : 0.85
F1-score : 0.85
ROC-AUC : 0.85

Sur le jeu de test 0.80102 en accuracy d'après Kaggle. Ce qui me permet de me classer 856/2291.

IX. Conclusion

Le modèle optimisé par TPOT nous a permis d'atteindre un très bon score en accuracy sur la compétition, ce qui est un indicateur de performance fiable, permettant même de rivaliser avec de très bons modèles optimisés manuellement.

Avantages à utiliser TPOT :

- Gain de temps important
- Ratio effort/résultat obtenu optimal
- Facile à utiliser pour les novices

Axes d'amélioration pour obtenir de meilleures performances :

- Feature engineering un peu plus poussé
- Temps de calcul supplémentaire
- Puissance de calcul supérieure