

Projet 8 Parcours Machine Learning

kaggle

1

Introduction

2

Analyse exploratoire

3

Feature engineering

4

Prétraitement des données

5

Implémentation du modèle

6

Conclusion

1. 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 prédire 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 des informations sur les passagers du vol, telles que l'âge, le patronyme, la destination ou encore les différentes dépenses effectuées à bord du vaisseau.



2. Analyse exploratoire



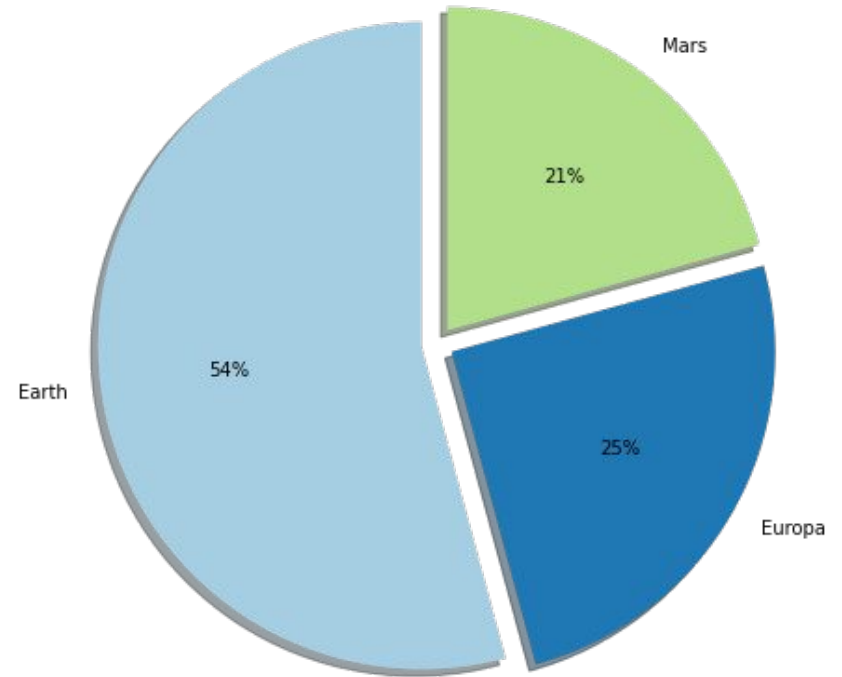
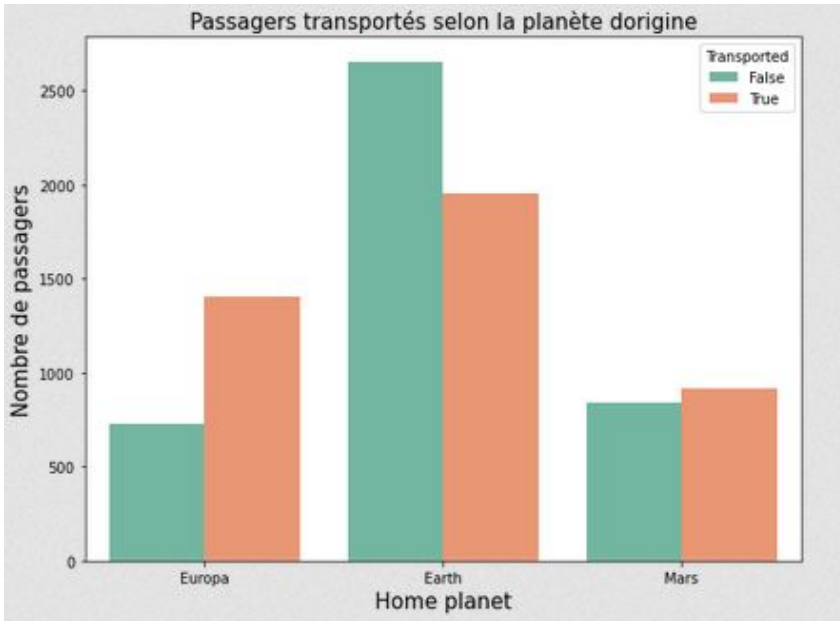
Les outils :

- Jupyter Notebook
- Kaggle (cloud computing)
- Bibliothèques de Data Science en python (Numpy, Pandas, Matplotlib, Seaborn)

Les données :

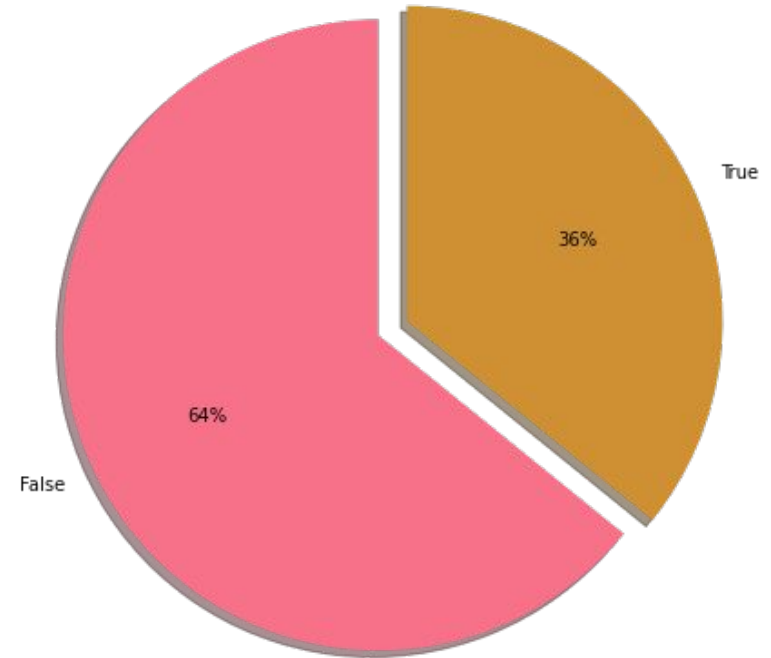
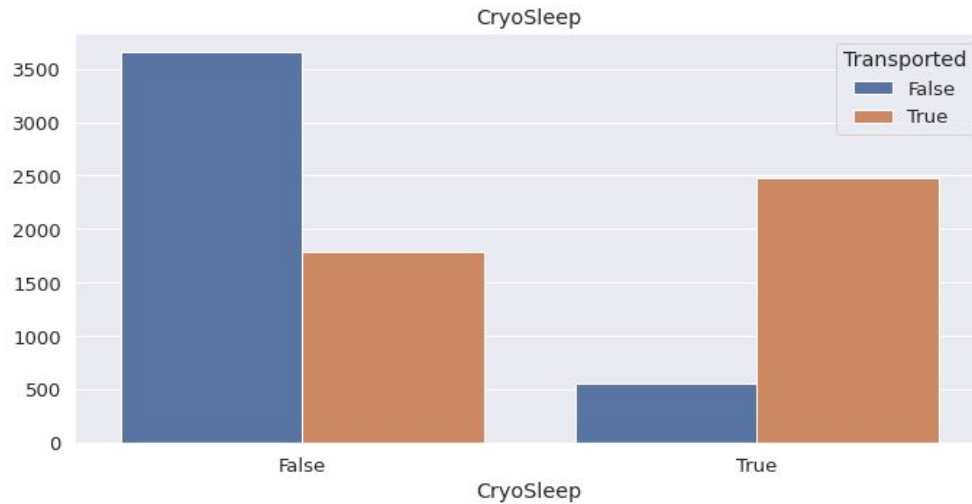
- Les données à disposition nous fournissent des informations sur les passagers et leur vol (Nom et prénom, planète d'embarquement, destination, dépenses effectuées...)
- Le jeu d'entraînement contient 8693 entrées tandis que le jeu de test en compte 4277.

Distribution des passagers selon la planète d'origine

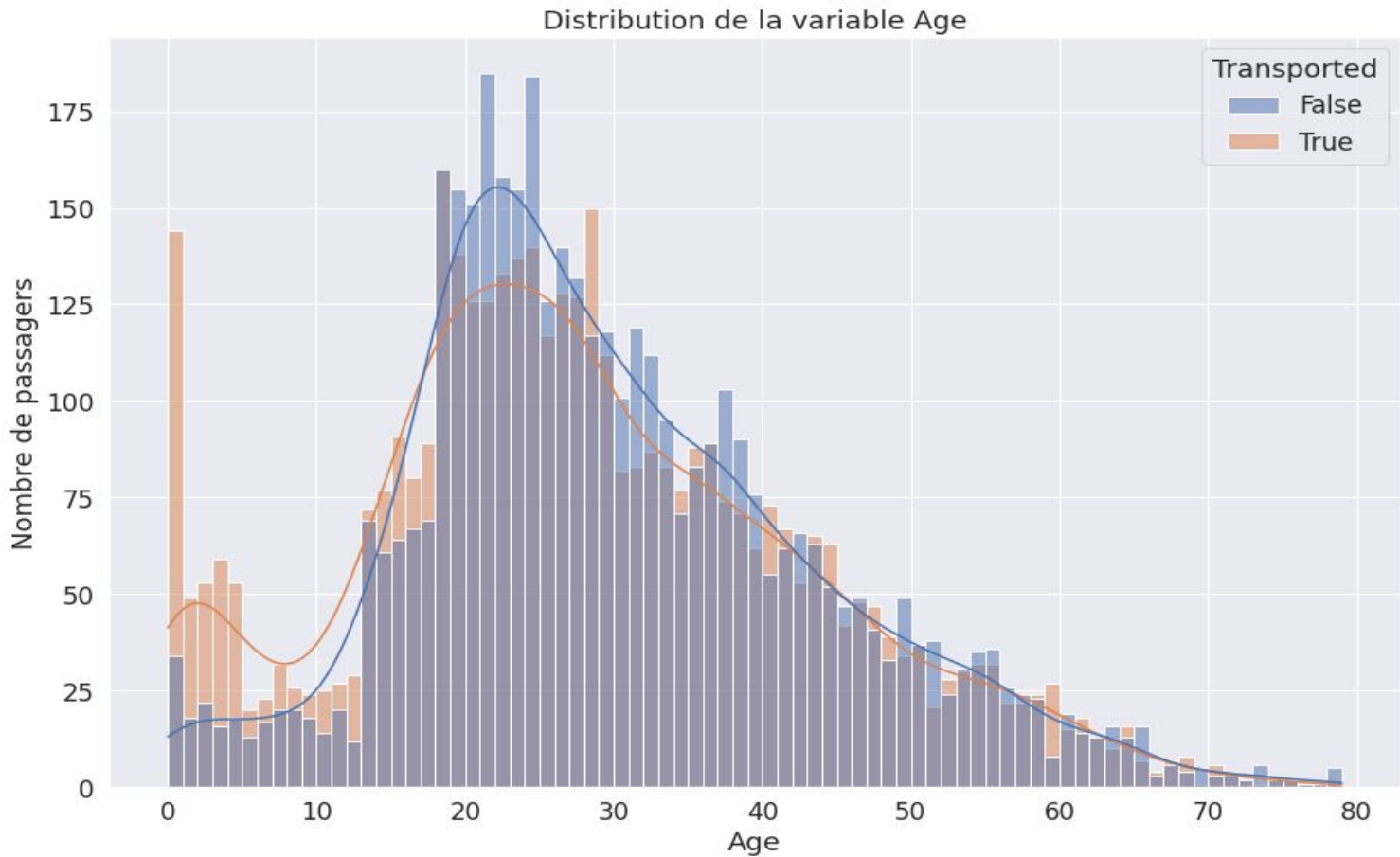


- Les voyageurs provenant de la terre ont plus de chances d'être transportés

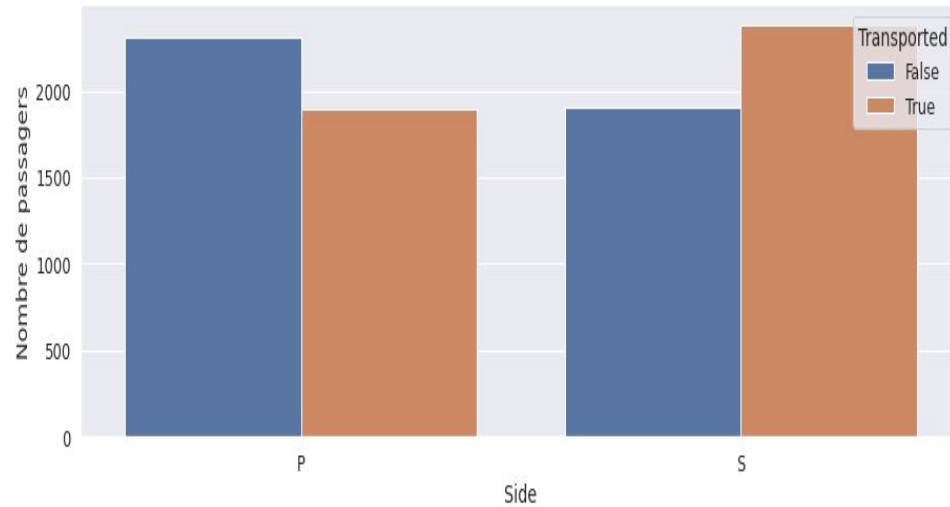
Distribution des passagers en Cryosleep



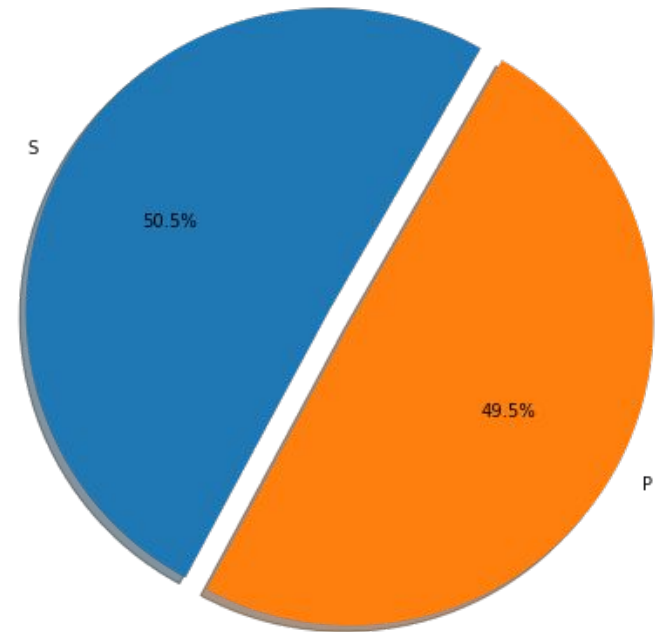
- Les voyageurs en sommeil cryogéniques sont favorisés dans le transport

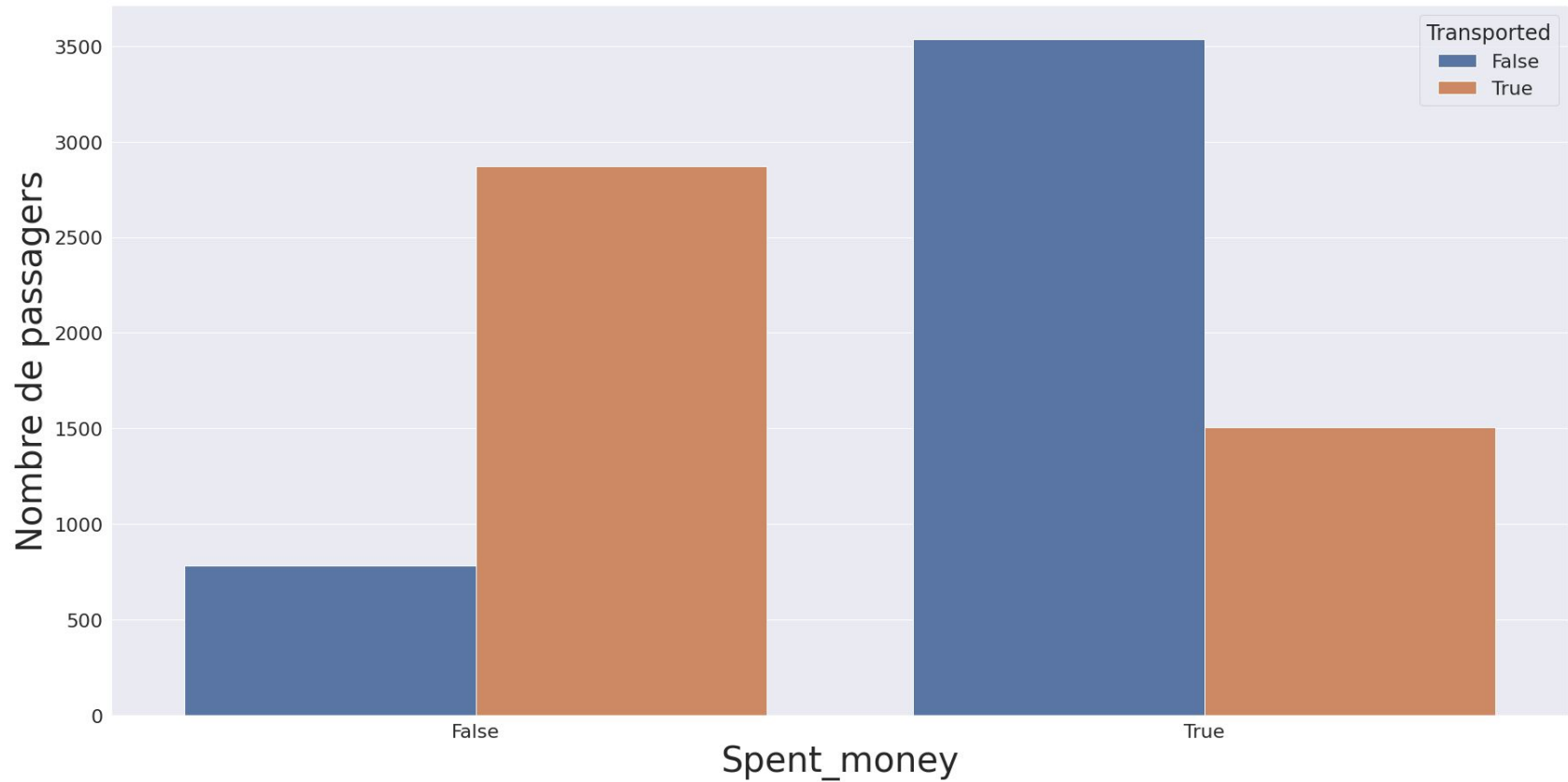


- Les voyageurs en bas âge sont plus susceptibles d'être transportés. Les jeunes adultes beaucoup moins.



Distribution des passagers dans les différents côtés des cabines





- Les voyageurs n'ayant rien dépensé ont nettement plus de chance d'être transporté.

3. 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
-

4. Prétraitement des données

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.

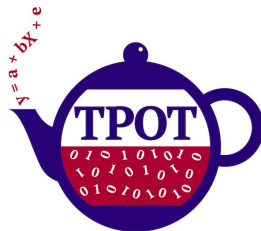
Features sélectionnées : "CryoSleep", "Spent_money", "Age", "RoomService", "FoodCourt", "ShoppingMall", "Spa", "VRDeck", "Total_expense", "HomePlanet", "Destination", "Deck", "Side".

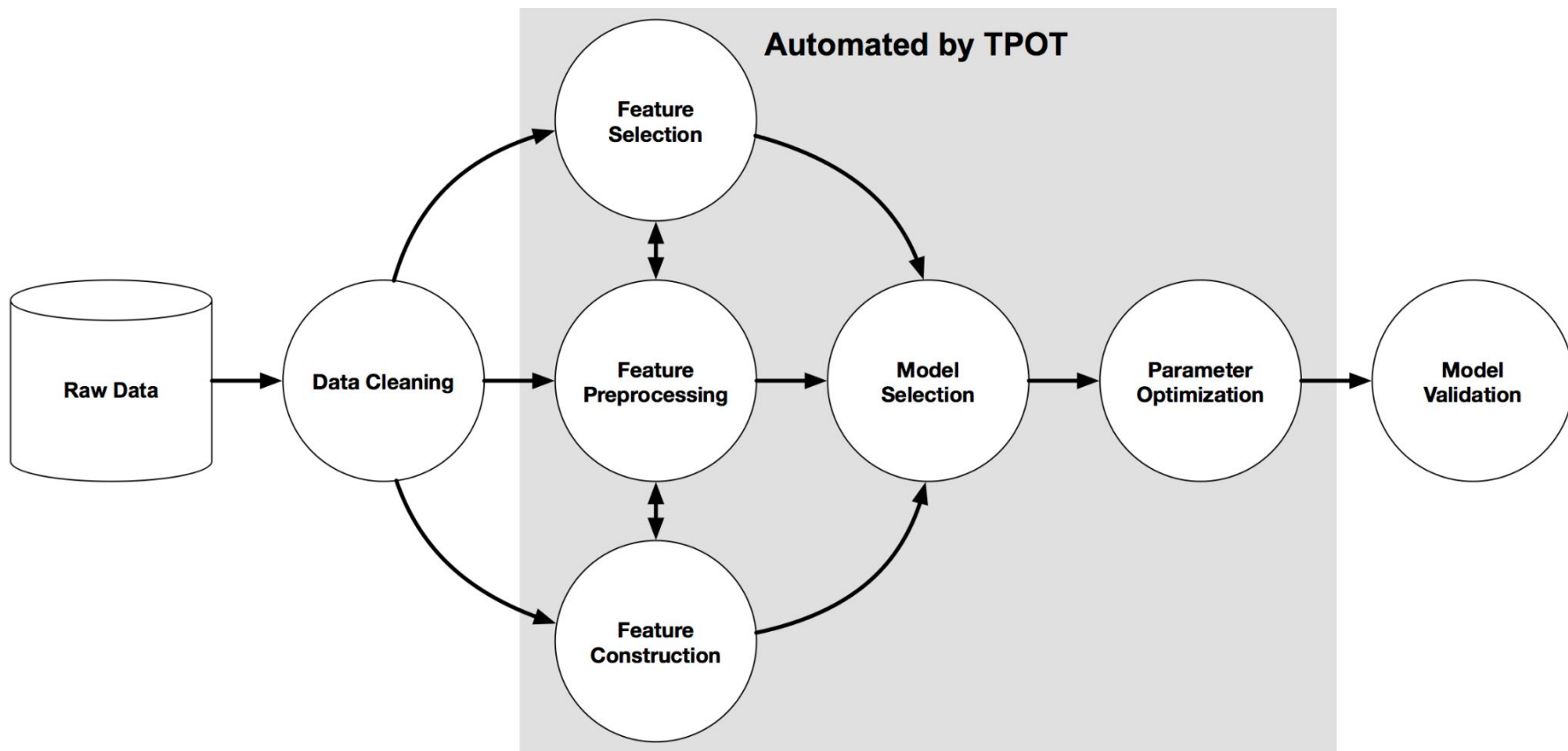
4. Implémentation du modèle TPOT

L'outil TPOT sera utilisé pour l'implémentation de notre modèle.

Il s'agit d'un algorithme génétique qui génère un nombre donné de modèles de façon aléatoire puis sélectionne les plus performantes d'entre eux afin d'opérer des crossovers (croisements) entre les modèles sélectionnés qui pourront, dans certains cas, s'avérer encore plus performants que les modèles originels.

L'avantage d'avoir recours à ce genre de procédé est d'automatiser le traitement Machine Learning





5. 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 on obtient 0.80102 en accuracy d'après Kaggle. Ce qui nous permet de nous classer 856/2291.

656	JOhnny Lerouge		0.80102	16	3d
-----	----------------	---	---------	----	----



Your Best Entry!
Your submission scored 0.79635, which is not an improvement of your previous score. Keep trying!

6. Conclusion (1 / 2)

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

6. Conclusion (2/2)

Axes d'amélioration pour obtenir de meilleures performances :

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

