Rapport

présenté et soutenu le : xx mai 2022 par

Aurélien CORROYER-DULMONT

dans le cadre de la formation

Ingénieur machine learning

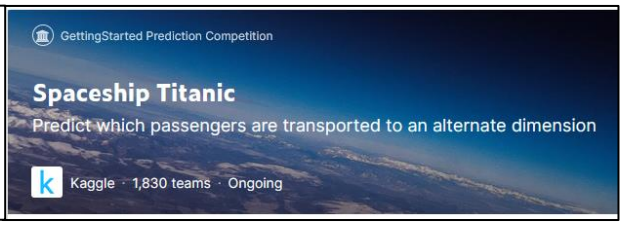
pour le

Projet n°8 « *Participez à une compétition Kaggle !* »

Openclassrooms

Formation effectuée au sein du :

Centre de lutte contre le cancer François Baclesse



I. Contexte et objectif

A. Appel à projet

Le site Kaggle propose des compétitions informatiques sur des sujets différents. Les missions de ce projet sont de :

- Participer à une de ces compétitions réelle et en cours
- Obtenir des résultats mesurables avec un classement
- Collaborer avec d'autres compétiteurs ou en équipe pour améliorer les modèles
- Présenter un notebook explicatif de la démarche pour participer à l'évolution collective

B. Compétition choisie et objectifs

Contexte : Un navire spatial comprenant 13 000 passagers a traversé une anomalie spatio-temporelle. L'ordinateur de bord d'un précédent voyage ayant connu la même fin nous informe de données de passagers ayant ou non était transporté dans la faille spatio-temporelle. Lien de la compétition : https://www.kaggle.com/competitions/spaceship-titanic/overview

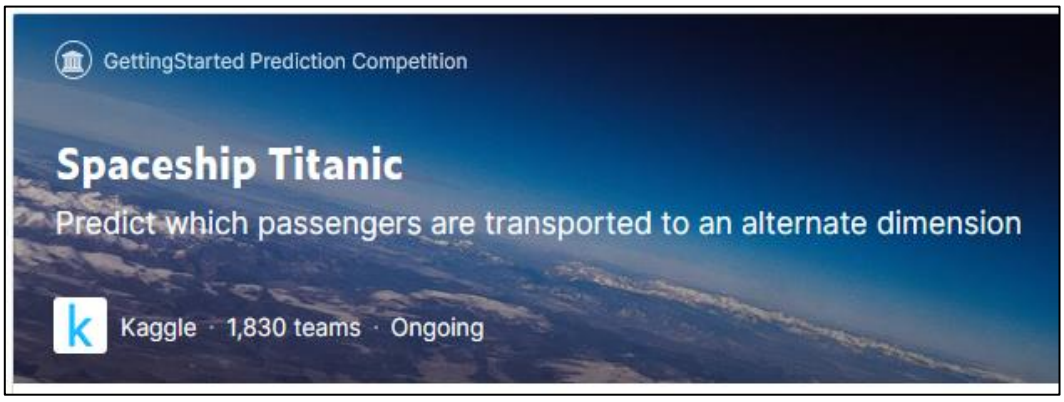


Figure 1 : Compétition Kaggle, titre et objectif

Objectif : A partir des informations de bord (nom des passagers, n° de cabin…), être capable de prédire avec des modèles de classification de machine learning ou de deep learning, si les passagers peuvent être sauvés ou non

II. Approche méthodologique

A. Nettoyage des données

Ce projet ne semblait pas contenir de valeurs aberrantes, je n'ai donc pas supprimer de valeurs. En revanche environ 25% des données contenaient des NaN, je les ai donc gérés de la manière suivante :

- **Planète d'origine / Destination** : mettre la planète la plus fréquente *(Europa* et *Trappist)*
- **Dépense totale** : si **cryosleep** = True alors mettre 0 sinon mettre la valeur moyenne
- **VIP** : mettre à *False* car sont les non **VIP** sont très majoritaire
- **Side/Deck** : mettre de façon aléatoire une lettre car les proportions sont homogènes
- **Cabin number** : mettre un chiffre aléatoire entre 1 et 1894
- **Cryosleep** : mettre la situation la plus fréquente *(False)*

B. Features engineering

A partir des données initiales j'ai pu créer six variables supplémentaires qui avaient selon moi une importance pour la modélisation.

- **Dépenses totales** *:* En utilisant la somme des variables de frais de service utilisés
- **FirstName** et **LastName** : en utilisant la variable **Name** (supposant qu'une famille à plus de probabilité de rester ensemble)
- **Deck**, **Side** et **Cabin number** : en utilisant la variable **Cabin** (car celle-ci regroupait plusieurs informations)

Tableau 1 : Variables utilisées pour la modalisation

Features quantitatifs	Features catégoriels
Age	Firstname
Cabin_number	Lastname
Dépense_totale	Home Planet
RoomService	Destination Planet
FoodCourt	VIP
ShoppingMall	Cryosleep
Spa	Deck

NB : pour la modélisation, j'ai utilisé un OneHotEncoder pour les variables catégorielles

C. Exploration des données

Cette partie avait pour objectif de comprendre le comportement des variables concernant la problématique afin de savoir lesquels seraient important pour la prédiction.

Proportion des passagers survivant ou non : il y a autant de passagers transportés que de non transportés **(Figure 2).** Il n'y aura donc pas de disproportion entre les deux classes à prédire.

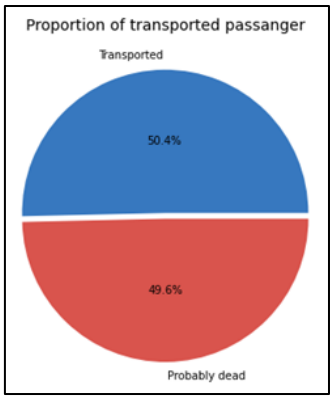


Figure 2 : Proportion de passagers transportés ou non

Importance de la classe des voyageurs (VIP ou non) : l'idée était ici de voir si le fait que les passagers soient VIP pouvait les amener à être plus transportés que la moyenne. La classe des VIP est très minoritaire et n'est pas avantageuse pour la problématique **(Figure 3).**

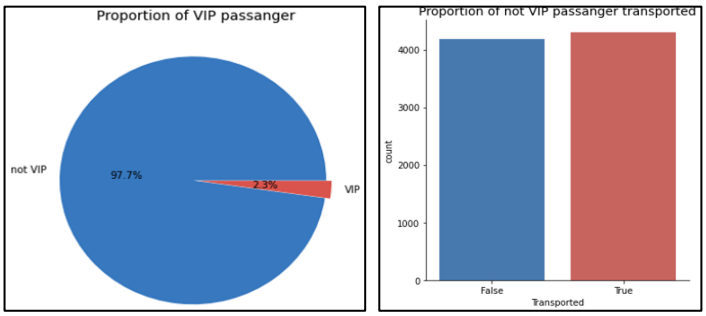


Figure 3 : Proportion des passagers VIP et non VIP et l'impact sur leurs survies

Importance de l'âge des passagers : Distribution assez homogène dans les adultes mais les plus jeunes ont cependant plus de chance de survivre **(Figure 4).**

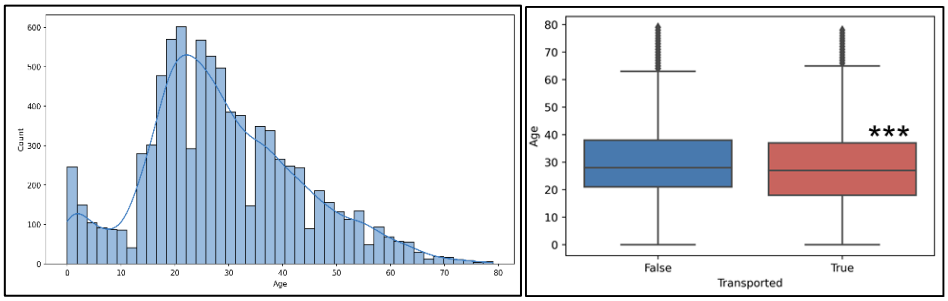


Figure 4 : Proportion des passagers VIP et non VIP et l'impact sur leurs survies

Critère sommeil cryogénique : Les personnes en sommeil cryogénique ont beaucoup plus survécu que les autres **Figure 5**.

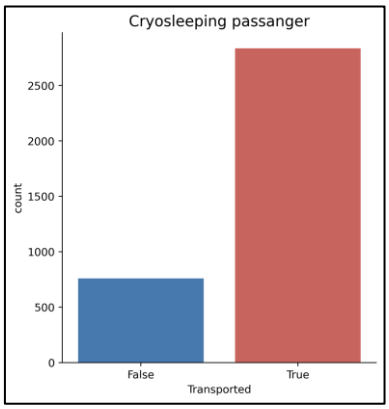


Figure 5 : Proportion des passagers en sommeil cryogénique transportés ou non

La planète d'origine a-t-elle un impact ? : Les passagers venant d'Europe semblent avoir été plus chanceux **(Figure 6).**

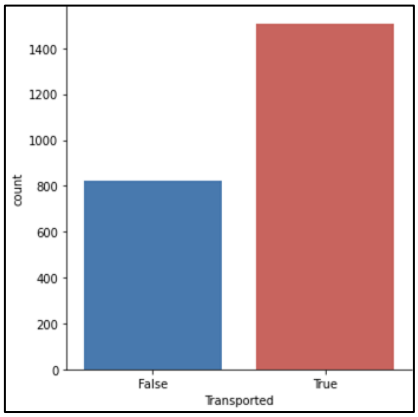


Figure 6 : Proportion des passagers venant d'Europe transportés ou non

La planète de destination a-t-elle un impact ? : Les passagers voyageant pour 55 Cancri e d'Europe semblent avoir été plus chanceux **(Figure 7).**

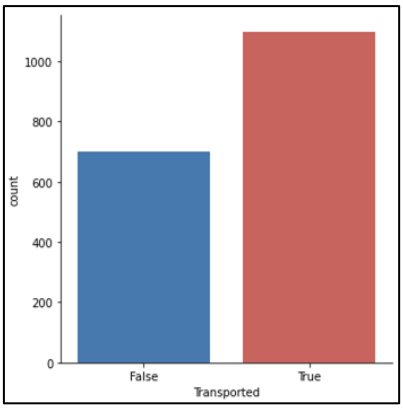


Figure 7 : Proportion des passagers voyageant pour 55 Cancri e transportés ou non

D. Algorithmes de modélisation

Pour cette modalisation j'ai choisi de tester différentes approches incluant des modèles de machine learning de classification ainsi que des modèles de deep learning :

Tableau 2 : algorithme de machine learning et deep learning utilisées pour la modalisation

Machine Learning	Deep Learning
LinearDiscriminant()	Tensorflow()
Ridge()	Pytorch()
KNeighbors()	
GaussianNB()	
RandomForestClassifier()	
SVC()	

III. Résultats

A. Modélisation machine learning

Les différents algorithmes mentionnés dans le **Tableau 2** ont été optimisé par recherche des meilleurs hyperparamètres par GridSearch avec validation croisée. Les performances des modèles ont dans un premier temps étaient comparées *via* le score d'entrainement et le score de prédiction. Les résultats sont visibles sur la **Figure 8** :

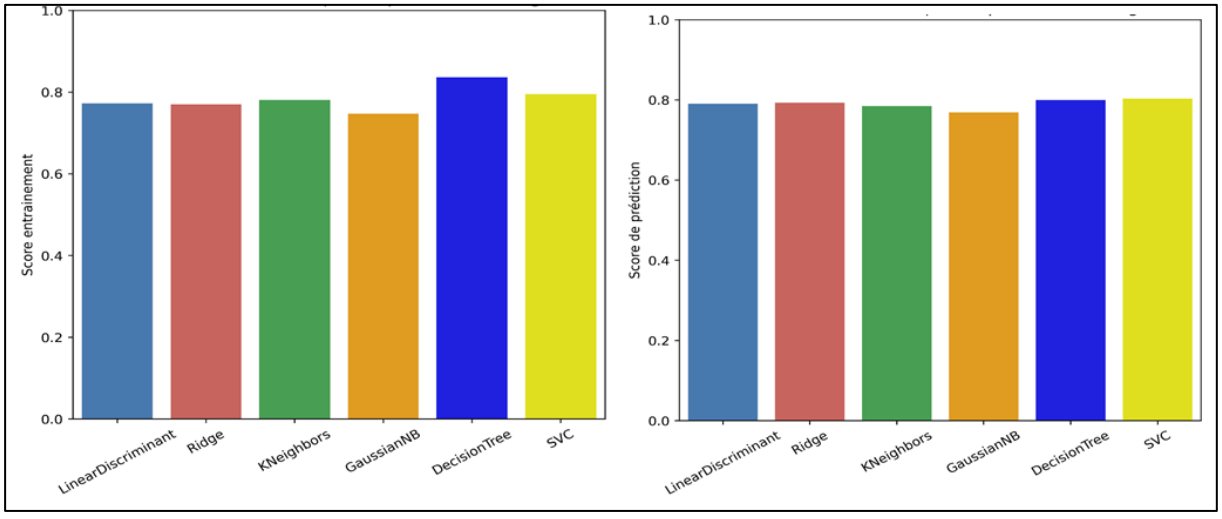


Figure 8 : Scores des modèles de machine learning

Les modèles RandomForestClassifier et SVC donnnent les meilleurs résultats avec une score de prédiction pour le SVC de 0.82.

B. Modélisation deep learning n°1

Le premier modèle de deep learning a utilisé la librairie tensorflow et keras. Voici son architecture :

- 1 couche de neurone d'entré (relu, 30 features)
- 2 couches de neurones cachés (relu, 782 neurones)
- 1 couche de sortie (softmax, 2 output de prédiction)

Les paramètres d'optimisation choisit ont été les suivants :

- **optimizer** = "*Nadam*"
- **loss** = "*BinaryCossentropy*"
- **metrics** = "*BinaryAccuracy*"
- 641 242 paramètres entrainés

Ce modèle permet un entrainement sans overfitting (**Figure 9**) mais avec un score de prédiction de 0.82.

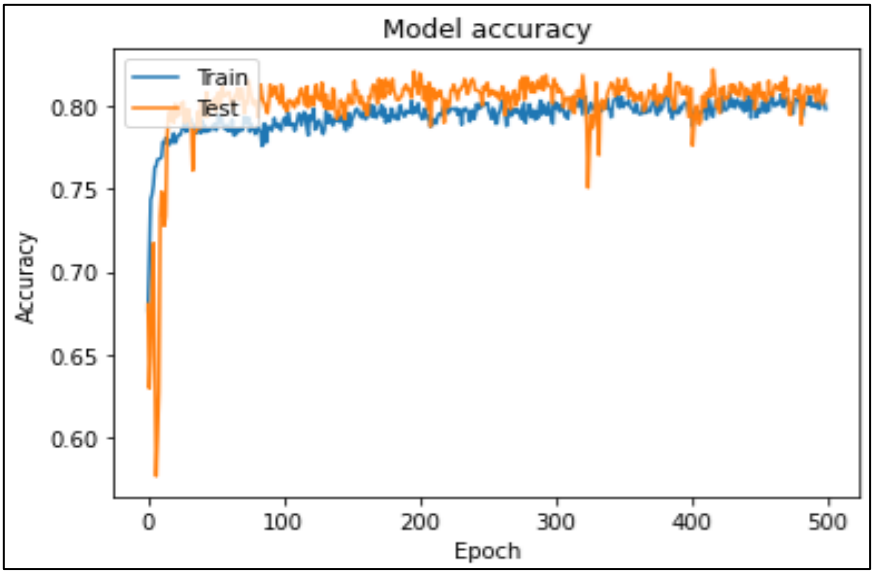


Figure 9 : Courbe d'apprentissage du modèle de deep learning n°1

C. Modélisation deep learning n°2

Le deuxième modèle de deep learning a consisté à utiliser la librairie keras-tuner pour essayer une optimisation des hyperparamètres du modèle de deep learning. Les hyperparamètres testés ont été les suivants :

- **Activation couches cachées :** "*elu, gelu, relu, selu*"
- **Activation couche sortie :** "*sigmoid, hard_sigmoid, softmax, swish, tanh*"
- **Range nb neurone** : "100=>len(X)/10" ;
- **Learning_rate :** "0.0005 à 0.1"

Les meilleurs hyperparamètres choisis ont été 300 et 500 neurones pour les deux couches de neurones cachés avec une activation de type relu et de type sigmoid pour la couche de sortie, enfin le learning rate optimum était de 0.05. Malgré cette optimisation, le score d'accuracy était de 0.80.

D. Modélisation deep learning n°3

Le modèle n°3 se différencie des autres par l'utilisation de pytorch plutôt que tensorflow avec une architecture de type multi-layer-perceptron :

- 1 couche de neurone d'entré (relu, 30 features)
- 3 couches de neurones cachés (relu, 256 neurones, dropout=0.2)
- 1 couche de sortie (relu, 1 valeur de prédiction)

Les paramètres d'optimisation choisit ont été les suivants :

- **optimizer** = "A*dam*"
- **loss** = "*BCEWithLogitsLoss*"
- **metrics** = "*BinaryAccuracy*"
- 135 809 paramètres entrainés

Ce modèle n°3 a permis d'obtenir un score d'accuracy sur le dataset de test de 0.82.

E. Prédiction et performance des modèles

Comme on peut le voir sur la **Figure 10** le modèle de machine learning n'est pas mauvais et dépasse même les deux premiers modèles de deep learning. Cependant le **modèle de deep learning n°3** de ce projet donne les meilleurs résultats avec **un score Kaggle de 0,79588.**

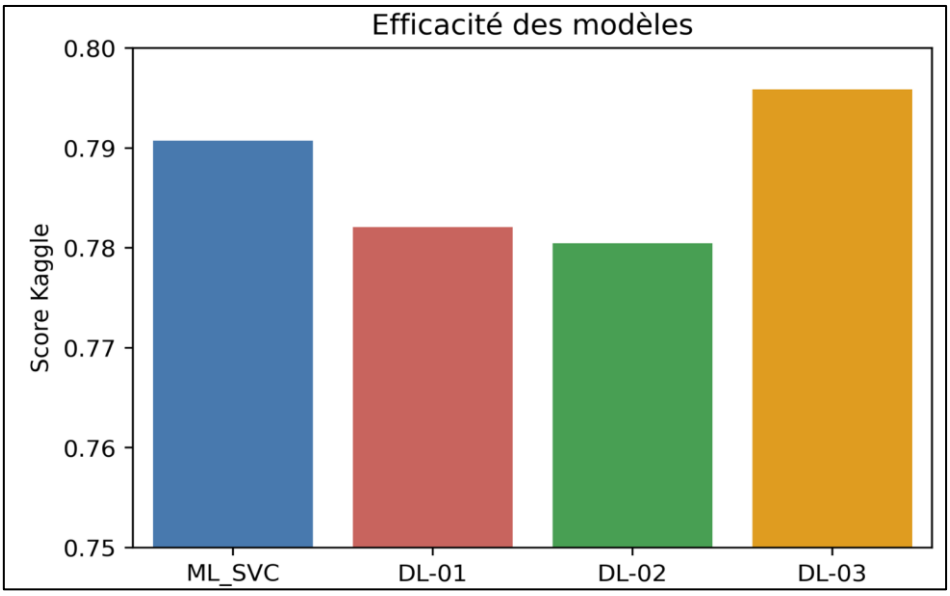


Figure 10 : Scores dans la compétition Kaggle en fonction des modèles étudiés

Ce score nous a permis de se placer (à la date du 04/05/2022) 788ème sur 1932 comme nous pouvons le voir sur la **Figure 11.**

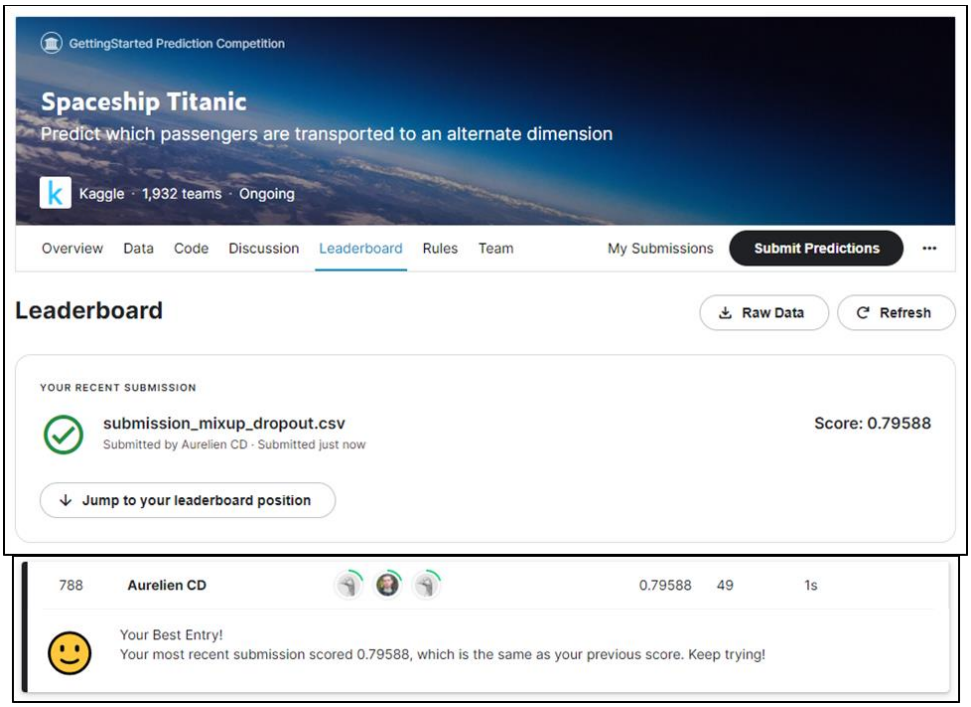


Figure 11 : Score final dans la compétition Kaggle

IV. Participer à l'évolution collective, présentation d'un notebook explicatif

Ce projet a été avant tout un travail d'équipe. Cyril Jaudet, PhD et physicien médical au centre de lutte contre le cancer François Baclesse ainsi qu'Ilyass Moummad, doctorant en informatique, m'ont aidé tout au long de ce projet, je les en remercie.

J'ai aussi publié le notebook expliquant notre démarche sur un kernel Kaggle pour discuter de notre approche avec d'autres personnes. Lien du kernel

Enfin j'ai essayé de faire avancer les projets d'autres participants en leur conseillant des approches. Lien du kernel

V. Conclusion

L'objectif de ce projet était de participer à une compétition informatique Kaggle ainsi que de travailler en collaboration en équipe ou de discuter avec d'autres participants. J'ai ainsi pu participer en équipe à un Kaggle de prédiction de classification où j'ai réussi à me classer 788ème sur 1982 avec comme modèle final choisi un modèle de deep learning MLP utilisant pytorch.