Travail d'étude et de recherche sur la modélisation et la prédiction de survie des individus face au naufrage du Titanic

Mulapi Tita Ketsia

Université de Rouen Normandie

contact@ketsiamulapi.com

January 17, 2022

Sommaire

Introduction

Le naufrage du Titanic a mis fin au paquebot qui devait relier Southampton à New York. Il se déroule dans la nuit du 14(22H) au 15 avril 1912 dans l'océan Atlantique Nord. (Wikipédia)

Ce travail, est principalement axé sur la prédiciton des survies des individus présents au moment des faits. Nous avons eu l'occasion de comprendre ce qui font ces individus à l'aide de leurs caractéristiques dont; le rang social, l'age, le sexe, etc. qui ont su avoir un impact sur la probabilité de survie.

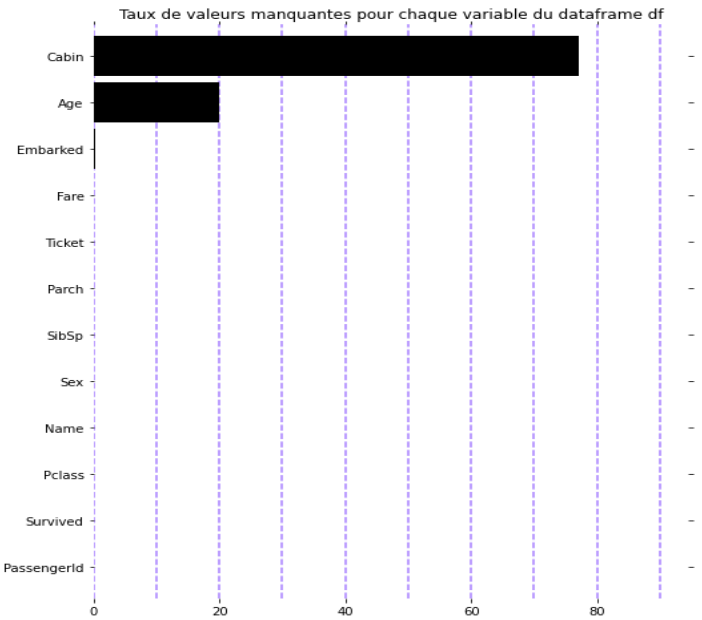
Dans la suite, 3 parties sont abordées, lesquelles seront suivies des résultats qui précède la conclusion, il s'agit notamment de :

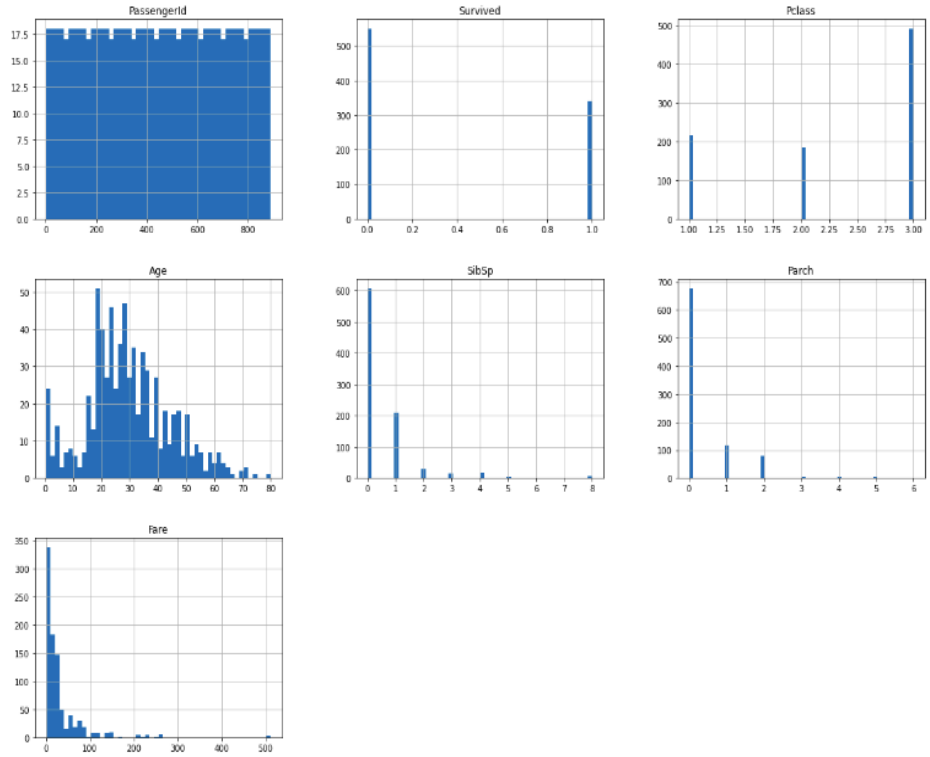
1. Exploration et compréhension des données
2. Approche de résolution Forte
3. Approche Faible pour l'explicabilité

Prétraitement : des données brutes aux prêtes à utiliser

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  418 non-null    int64
 1   Pclass       418 non-null    int64
 2   Name         418 non-null    object
 3   Sex          418 non-null    object
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64
 6   Parch        418 non-null    int64
 7   Ticket       418 non-null    object
 8   Fare         417 non-null    float64
 9   Cabin        91 non-null     object
 10  Embarked     418 non-null    object
dtypes: float64(2), int64(4), object(5)
memory usage: 36.0+ KB
```

	Survived	PassengerId	Pclass	Age	Sex_F	Sex_M	SibSp	Parch	Fare	Embarked_C	Embarked_Q	Embar
count	889.000000	889.000000	889.000000	889.000000	889.000000	889.000000	889.000000	889.000000	889.000000	889.000000	889.000000	889.0
mean	0.382452	446.000000	2.311586	29.278403	0.350956	0.649044	0.524184	0.382452	32.096681	0.188976	0.086614	0.7
std	0.486260	256.998173	0.834700	14.199591	0.477538	0.477538	1.103705	0.806761	49.697504	0.391710	0.281427	0.4
min	0.000000	1.000000	1.000000	0.420000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
25%	0.000000	224.000000	2.000000	20.000000	0.000000	0.000000	0.000000	0.000000	7.895800	0.000000	0.000000	0.0
50%	0.000000	446.000000	3.000000	28.000000	0.000000	1.000000	0.000000	0.000000	14.454200	0.000000	0.000000	1.0
75%	1.000000	668.000000	3.000000	37.000000	1.000000	1.000000	1.000000	0.000000	31.000000	0.000000	0.000000	1.0
max	1.000000	891.000000	3.000000	80.000000	1.000000	1.000000	8.000000	6.000000	512.329200	1.000000	1.000000	1.0





Les aglorithmes pour la modélisation et la classification des données

Une synthèse de notre expérinece sur les algorithmes utilisés

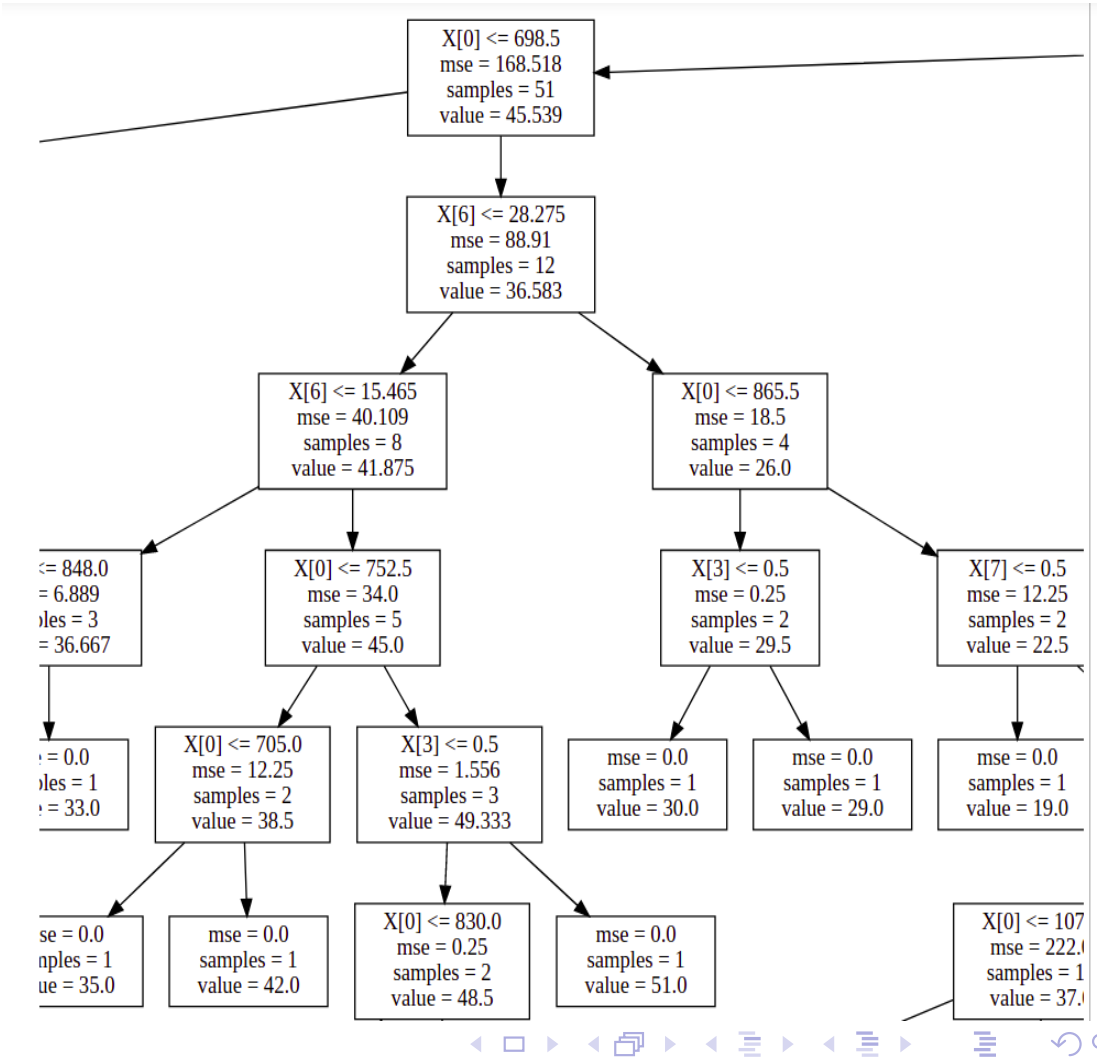
Nom	**Objectif-Modèle**	**Observation**
Decision Tree Regressor	Prediction	Inférence sur les ages
Xgboost	Classification	sur apprentissage
Random Forest Classifier	Classification	Satisfait
Decision Tree Classifier	Classification	Satisfait
SVM	Classification	3 SVC
Logistic Regression	Classification	Pas si bon que ça
Multi Layer Perceptron	Classification	Pas bien configuré
Gaussian	Classification	Pas adapté
SGDR	Classification	Pas bien configuré
KNN	Classification	Pas adapté

Table: Les algorithmes

Les aglorithmes pour la modélisation et la classification des données

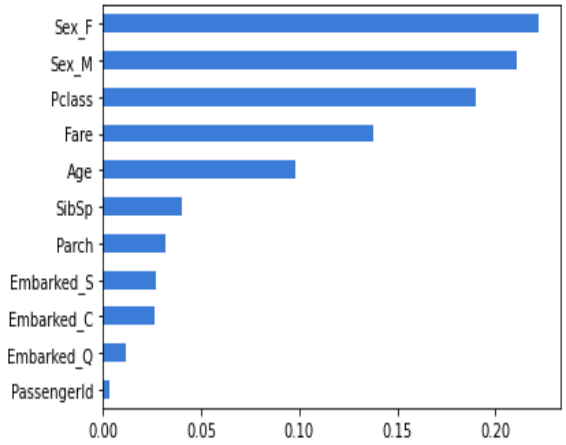
L'optimisation de notre SVM (type-SVC) et la prédiction des âges.

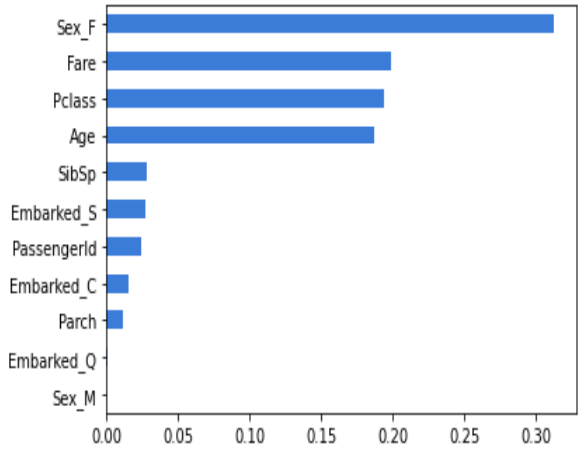
```
# Tuning hyper-parameters for precision

Best parameters set found on development set:

{'C': 10, 'kernel': 'linear'}

Grid scores on development set:

0.546 (+/-0.273) for {'C': 1, 'gamma': 0.001, 'kernel': 'rbf'}
0.637 (+/-0.212) for {'C': 1, 'gamma': 0.0001, 'kernel': 'rbf'}
0.522 (+/-0.216) for {'C': 10, 'gamma': 0.001, 'kernel': 'rbf'}
0.636 (+/-0.292) for {'C': 10, 'gamma': 0.0001, 'kernel': 'rbf'}
0.572 (+/-0.319) for {'C': 100, 'gamma': 0.001, 'kernel': 'rbf'}
0.544 (+/-0.145) for {'C': 100, 'gamma': 0.0001, 'kernel': 'rbf'}
0.563 (+/-0.333) for {'C': 1000, 'gamma': 0.001, 'kernel': 'rbf'}
0.605 (+/-0.183) for {'C': 1000, 'gamma': 0.0001, 'kernel': 'rbf'}
0.774 (+/-0.038) for {'C': 1, 'kernel': 'linear'}
0.803 (+/-0.081) for {'C': 10, 'kernel': 'linear'}
0.801 (+/-0.066) for {'C': 100, 'kernel': 'linear'}
0.793 (+/-0.069) for {'C': 1000, 'kernel': 'linear'}

# Tuning hyper-parameters for recall

Best parameters set found on development set:

{'C': 10, 'kernel': 'linear'}

Grid scores on development set:

0.486 (+/-0.071) for {'C': 1, 'gamma': 0.001, 'kernel': 'rbf'}
0.563 (+/-0.118) for {'C': 1, 'gamma': 0.0001, 'kernel': 'rbf'}
0.485 (+/-0.065) for {'C': 10, 'gamma': 0.001, 'kernel': 'rbf'}
0.512 (+/-0.028) for {'C': 10, 'gamma': 0.0001, 'kernel': 'rbf'}
0.473 (+/-0.079) for {'C': 100, 'gamma': 0.001, 'kernel': 'rbf'}
0.526 (+/-0.140) for {'C': 100, 'gamma': 0.0001, 'kernel': 'rbf'}
0.467 (+/-0.090) for {'C': 1000, 'gamma': 0.001, 'kernel': 'rbf'}
0.546 (+/-0.116) for {'C': 1000, 'gamma': 0.0001, 'kernel': 'rbf'}
0.762 (+/-0.045) for {'C': 1, 'kernel': 'linear'}
0.774 (+/-0.036) for {'C': 10, 'kernel': 'linear'}
0.768 (+/-0.034) for {'C': 100, 'kernel': 'linear'}
0.774 (+/-0.035) for {'C': 1000, 'kernel': 'linear'}
```

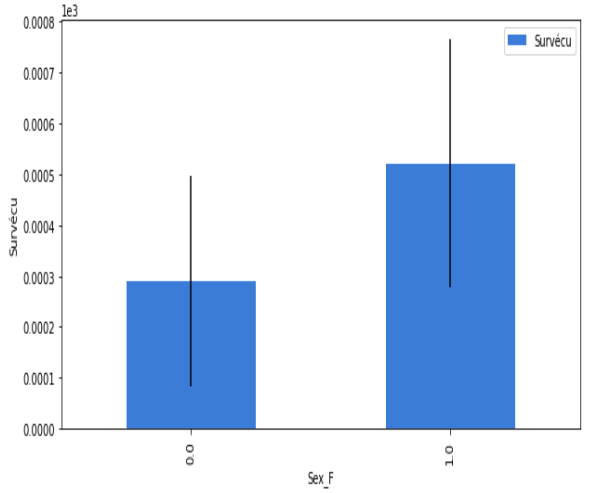


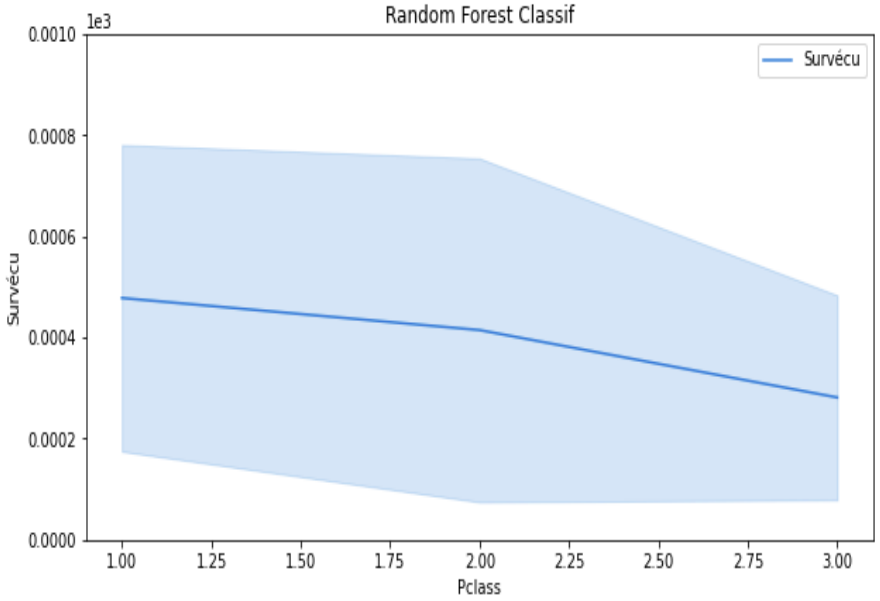
boîtes noires : les informations cachées avec Skater

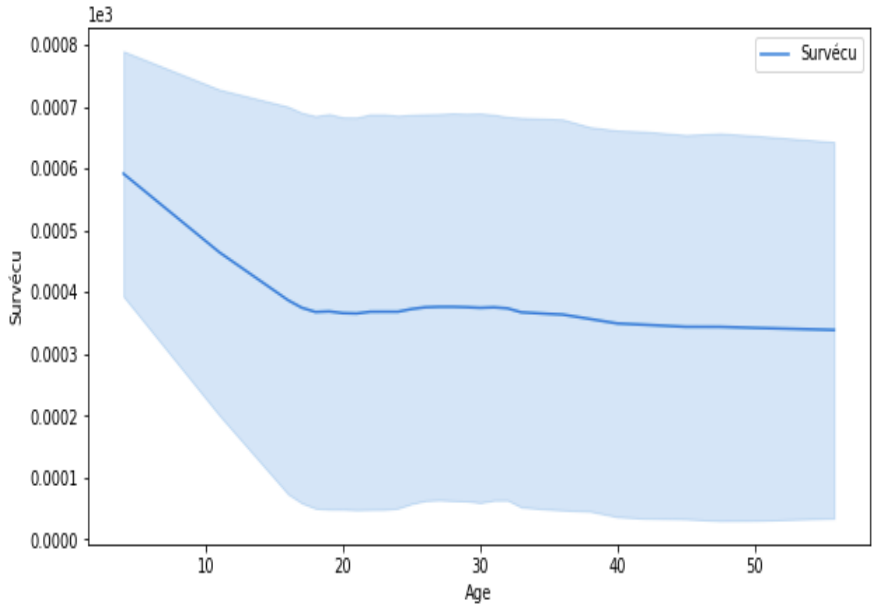
Basées sur un RandomForest(1,3,4,5) et un Xgboost(2).





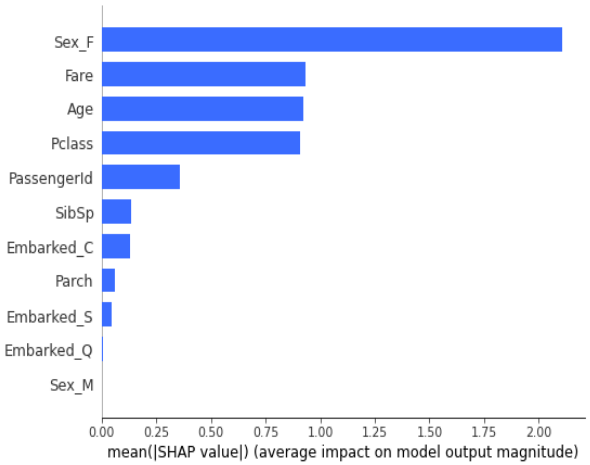


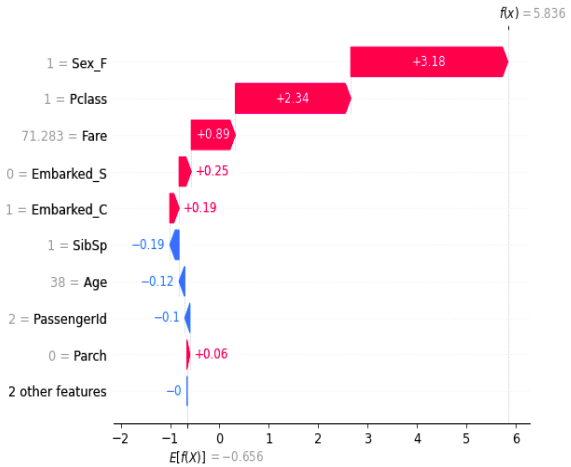


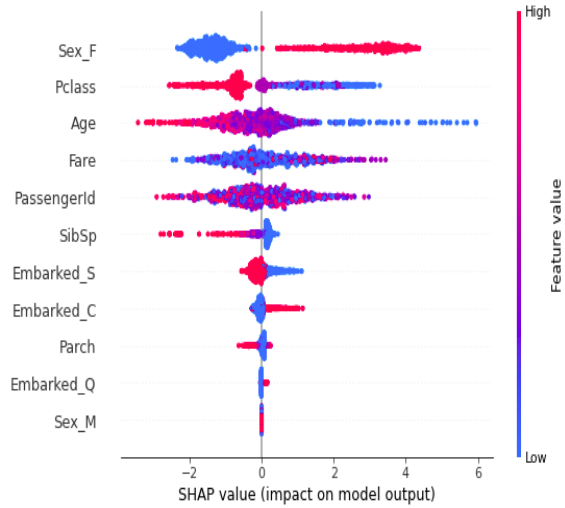


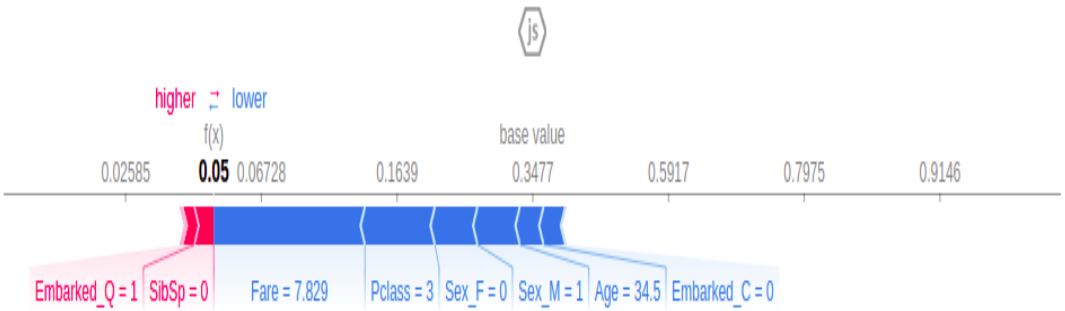
boîtes noires : les informations cachées avec Shape

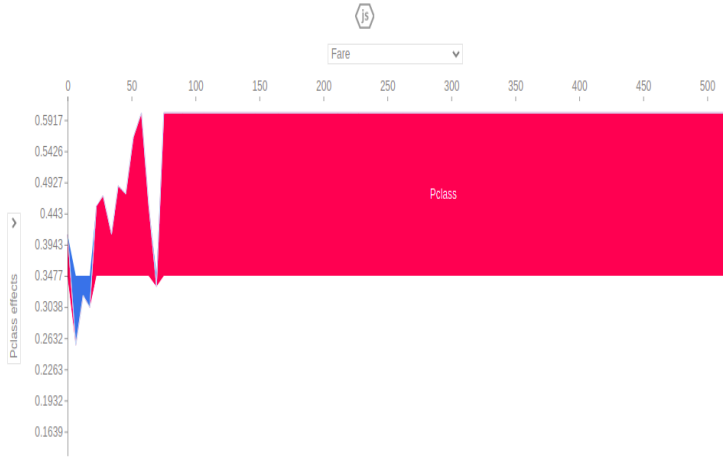
Dans l'ordre : Xgboost(1,2,3), Regression Logistique(4,5)











Résultat et conclusion

Résumé

Au temre de ce travail nous avons su prédire si un individu a survécu ou pas. La méthodologie employée commence par une phase de prétraitement, suivit du choix des modèles dont les explications ont été renforcées en faisant de l'explicabilité tout juste avant de prédire les résultat lors de la soumission sur kaggle. De ce fait, nous avons réalisé le meilleur score de 77.272% à l'aide d'un RandomForest, ce qui nous a emmené à la 8222e position comme vous pouvez le voir dans les fichiers LeaderBoard et Top5.