

Le processus du modèle a été intégré dans un script python: changez la liste des features souhaitées et lancez le script pour obtenir des résultats directement exportés dans un dossier img/.

Tester des combinaisons de variables !

\1. Choisissez vos features

Faites des test rapides en modifiant seulement la liste des features souhaitées: ouvrez le fichier **ML/Model.py** et modifiez la liste des features dans la variable self.features:

```
10 class Model:
11     def __init__(self):
12         self.features = ['Sex']
13         self.model = LogisticRegression()
```

```
10 class Model:
11     def __init__(self):
12         self.features = ['Sex', 'Pclass']
13         self.model = LogisticRegression()
```

2. Faîtes tourner le modèle

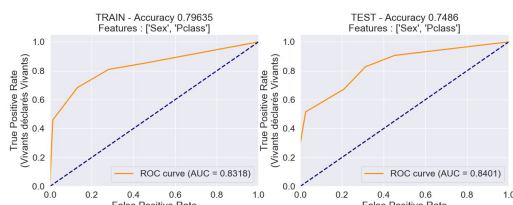
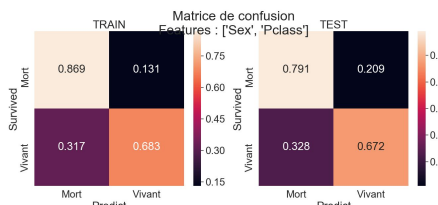
Exécutez le script **regression_logistic.py**

[C:\Emplacemen\du\Fichier > python regression_logistic.py]

```
C:\Users\Administrateur\Documents\simplon\python\Projet groupe\titanic> python .\regression_logistic.py
```

3. Visualisez vos résultats

Une courbe roc et une matrice de confusion ont été générées dans le dossier /img, elles contiennent les comparaisons du test et du train.



Et si tout marche bien, des barplots indiquants les erreurs sont générés en bonus !

OK, mais les features que je veux ne sont pas implémentées dans le modèle

Les étapes du modèle ont été fragmentées afin de pouvoir ajouter simplement de nouveaux traitements , Il y a 3 fichiers qui contiennent les fonctions de traitement de données, dans le dossier ML/:

- **TransformData.py** *Encoding features, changements de type*
- **NaTreatment.py** *Remplacement de données manquantes*
- **CreateFeature.py** *Extraction de nouvelles colonnes*

Modifiez le comportement du traitement d'une feature, ou ajoutez de nouvelles fonctions.

Pour activer/désactiver une fonction dans le traitement de données, rendez-vous dans le fichier **ProcessData.py**.

Exemple/Tuto :

Je souhaite ajouter une nouvelle colonne "AgeClass" contenant des groupes correspondants aux tranches d'âges.

Comme c'est une création de colonne, j'ouvre le fichier **ML/CreateFeature.py**

```
CreateFeature.py
1 import pandas as pd
2 import numpy as np
3 #
4 # --- CREATE FEATURES -----
5
```

J'ajoute une nouvelle fonction avec un nom explicite, avec comme paramètre un dataframe:

```
1 import pandas as pd
2 import numpy as np
3 #
4 # --- CREATE FEATURES -----
5 def extract_age_class(df):
6     df['AgeClass'] = np.digitize(df['Age'], bins=[1,6,16,26,31,37,42,100])
7
```

J'ajoute ma fonction au traitement général des données dans le fichier **ProcessData.py**. Utilisez le nom du fichier où la fonction a été écrite pour appeler votre nouvelle fonction.

```

5  def processRegressionLogistic(df):
6
7      # -----
8      # --- TRAITEMENTS DES DONNEES MANQUANTES -----
9      print('remplacement des données manquantes..')
10     NaTreatment.treatment_na_embarked(df)
11     NaTreatment.treatment_na_age(df)
12     # -----
13     # --- CREATE FEATURES -----
14     print('Création de Features..')
15     CreateFeature.extract_age_class(df)
16
17     # -----
18     # --- TRANSFORMATION DE FEATURES -----
19     print('features encoding..')
20     TransformData.transform_sex(df)
21     TransformData.transform_embarked(df)

```

Je souhaite ajouter ma nouvelle colonne à la liste des features pour mon modèle

```

10  class Model:
11      def __init__(self):
12          self.features = ['Sex', 'Pclass', 'AgeClass']
13          self.model = LogisticRegression()

```

Il ne me reste plus qu'à faire tourner le script et visualiser mes nouveaux résultats.