Université Paris Cité

Master 1 Machine Learning pour la Science des Données

Application Web

pour l'apprentissage supervisé pour les problèmes

de classification avec R SHINY

Réalisé par: Amira TLATI, Wessal HAMZA, Naïla KADDOURI

Année universitaire : 2021-2022

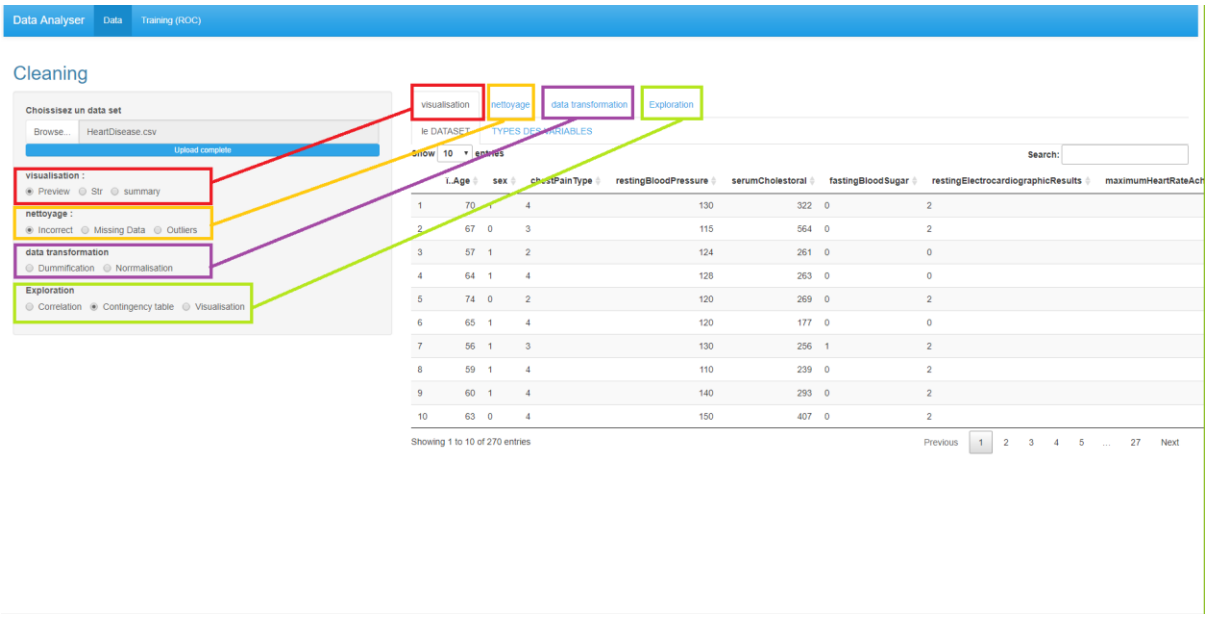
SOMMAIRE

1. Description générale de l'application :

Notre application web est constituée de 2 onglets, l'onglet data et l'onglet Training.
Dans le 1er on a un side panel, qui contient les sections suivantes :

- **choix du DataSet:** il s'agit d'un FileInput permettant de choisir un fichier .csv contenant notre dataset.
- **Visualisation des données**: l'utilisateur pourra parcourir son Dataset, voir les attributs présents et leur types mais aussi modifier le type de chaque variable.
- **Nettoyage des données:** inclut la gestion des valeurs manquantes, la suppression des outliers, la normalisation et la dummification
- **Exploration des données**: permet la visualisation, la corrélation des différentes variables selon le choix de l'utilisateur, l'affichage de la table de contingence et la visualisation (affichage de diagramme en bâton et de nuage de points).

Le main panel se met à jour selon l'option choisie par l'utilisateur comme présenté ci-dessous:



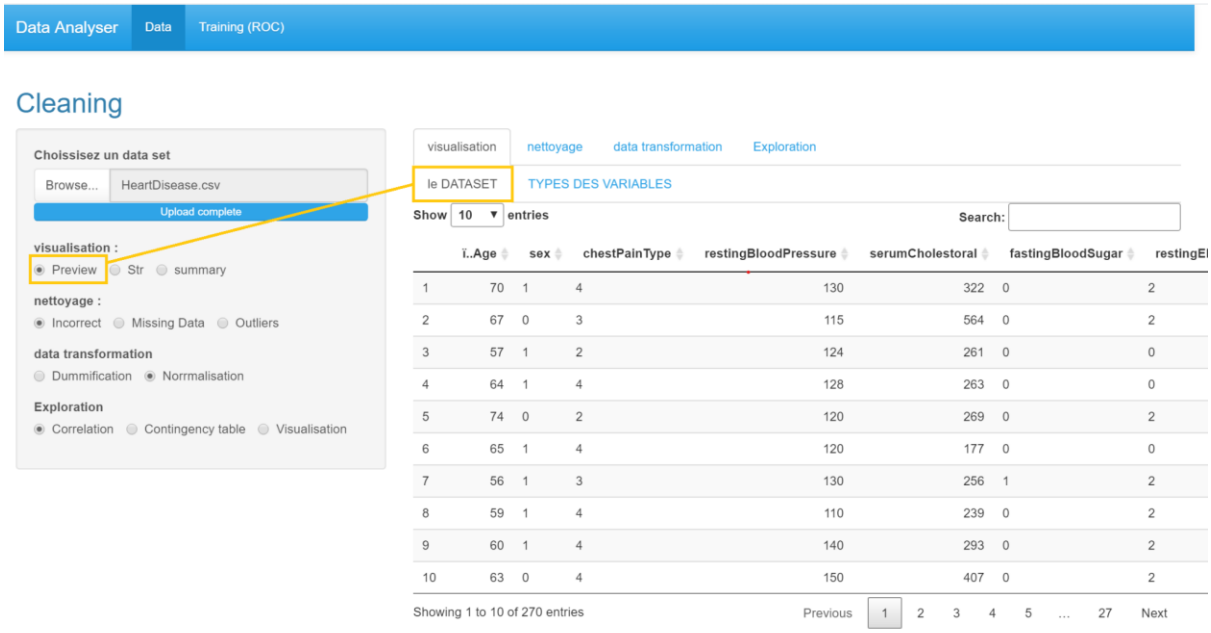
Le 2ème onglet contient trois modèles de classification supervisés et le calculs des différentes métriques correspondant à chaque modèle.

2. Mise en place de l'application :

Nous allons d'abord nous concentrer sur la mise en page, qui est définie dans l'objet ui (user interface). Premièrement, nous allons définir ui en tant que fluidPage, nous voulions diviser notre application en deux parties pour cela nous avons utilisé navbarPage. Cela créera une en-tête en haut de la page contenant Data et Training.

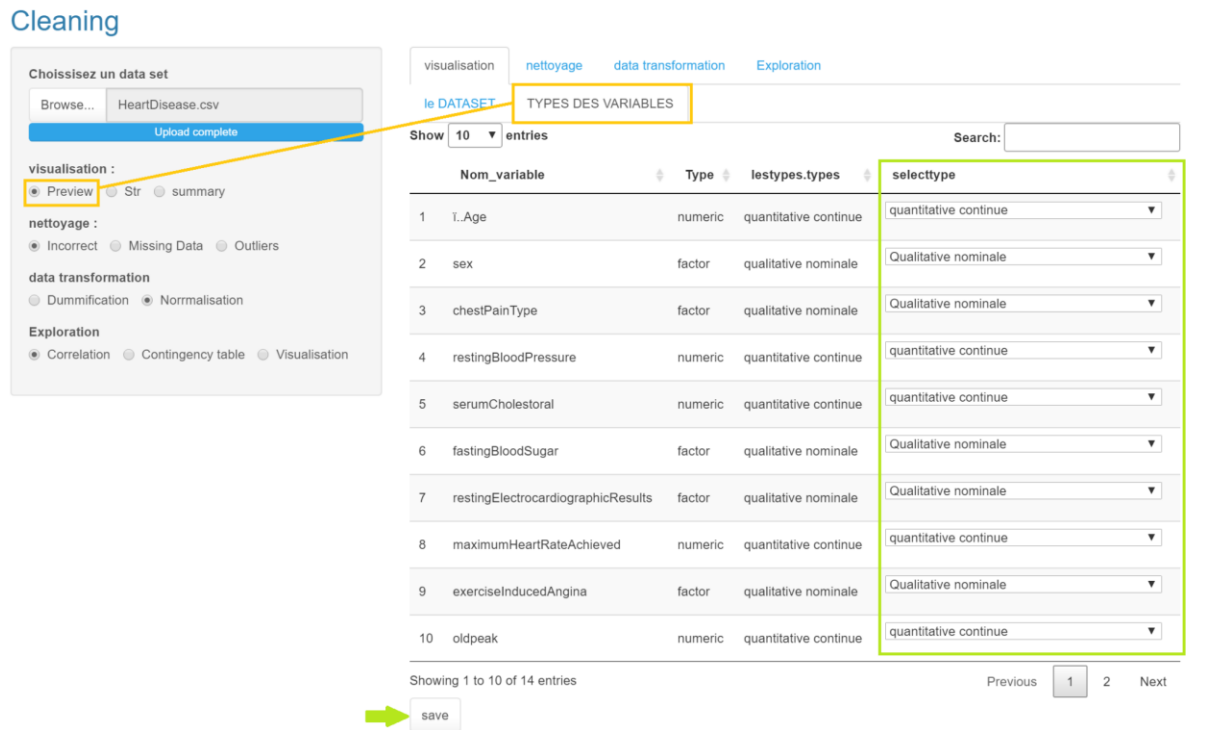
2.1. Visualisation

La partie visualisation permet premièrement de voir l'ensemble des données du dataset et cela en choisissant l'option Preview et en choisissant le panel DATASET.



En basculant vers le panel TYPES DES VARIABLES l'utilisateur aura la possibilité de visualiser et de modifier le type de chaque variable grâce à une liste déroulante qui contient l'ensemble des types possibles (quantitative discrète, quantitative continue, qualitative nominale, qualitative ordinale).

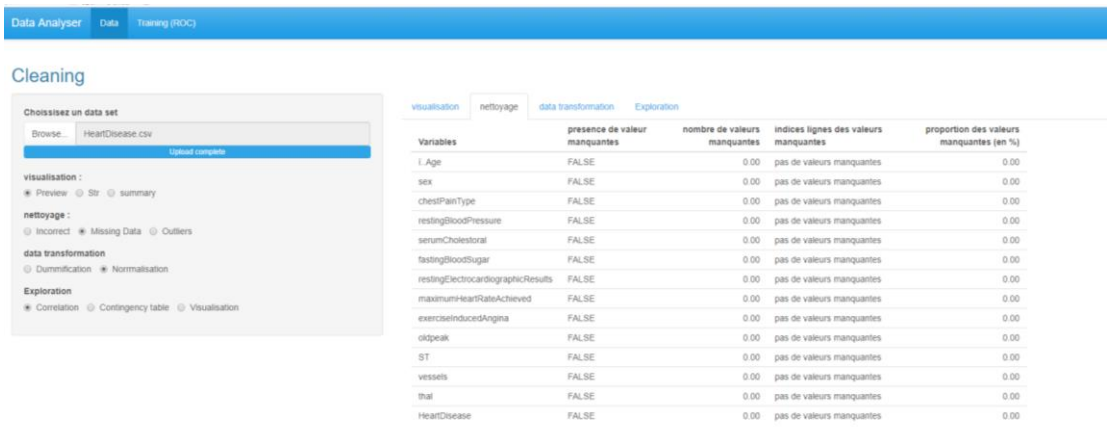
L'utilisateur devra cliquer sur le bouton "save" afin que ses changements prennent effet.



2-2. Nettoyage

2-2-1-Valeurs manquantes

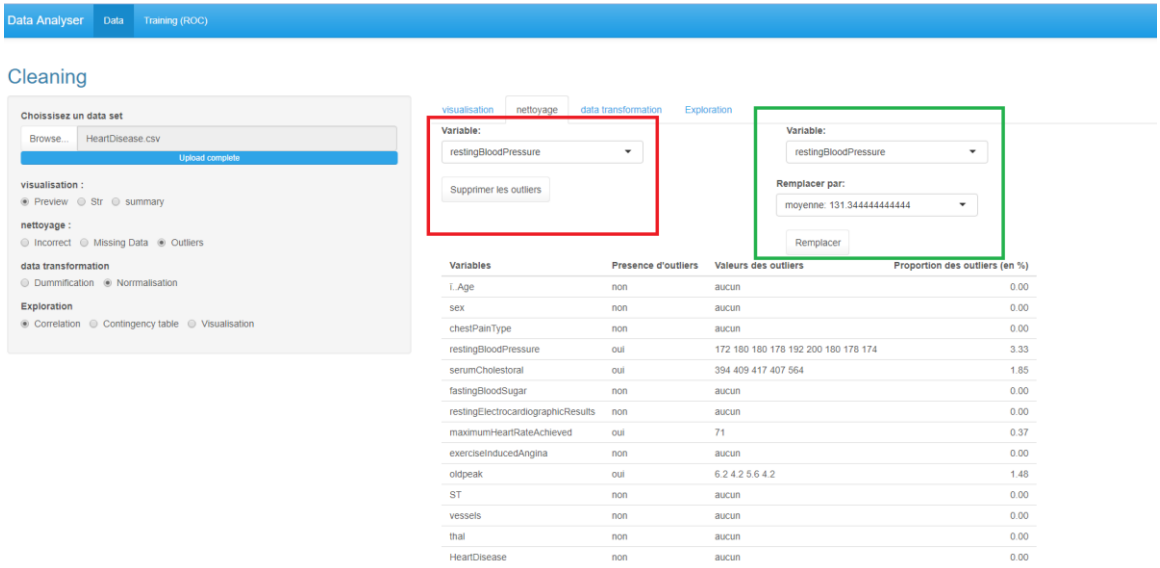
En appuyant sur le bouton radio “Missing data” et en allant sur l’onglet “nettoyage” notre application permet de supprimer les lignes qui contiennent des valeurs manquantes ou de les remplacer et d’afficher la liste des variables du dataset en précisant s' il y a la présence de valeurs manquantes et le pourcentage de celles-ci. Si il y a la présence de valeurs manquantes, des boutons select s’affiche juste en haut du dataframe qui liste les variables avec leur pourcentage de valeurs manquantes (voir exemple dans la partie 2-2-2- outliers) sinon aucun bouton ne s’affiche. Ainsi, cela signifie dans la capture ci-dessous que notre jeu de données ne contient aucune valeur manquante.



Variables	presence de valeur manquantes	nombre de valeurs manquantes	indices lignes des valeurs manquantes	proportion des valeurs manquantes (en %)
ï..Age	FALSE	0.00	pas de valeurs manquantes	0.00
sex	FALSE	0.00	pas de valeurs manquantes	0.00
chestPainType	FALSE	0.00	pas de valeurs manquantes	0.00
restingBloodPressure	FALSE	0.00	pas de valeurs manquantes	0.00
serumCholestoral	FALSE	0.00	pas de valeurs manquantes	0.00
fastingBloodSugar	FALSE	0.00	pas de valeurs manquantes	0.00
restingElectrocardiographicResults	FALSE	0.00	pas de valeurs manquantes	0.00
maximumHeartRateAchieved	FALSE	0.00	pas de valeurs manquantes	0.00
exerciseInducedAngina	FALSE	0.00	pas de valeurs manquantes	0.00
oldpeak	FALSE	0.00	pas de valeurs manquantes	0.00
ST	FALSE	0.00	pas de valeurs manquantes	0.00
vessels	FALSE	0.00	pas de valeurs manquantes	0.00
thal	FALSE	0.00	pas de valeurs manquantes	0.00
HeartDisease	FALSE	0.00	pas de valeurs manquantes	0.00

2-2-2-Outliers

L’application permet également d’afficher, de remplacer ou de supprimer les lignes qui contiennent des outliers (le principe est le même que pour les valeurs manquantes).



Variables	Presence d'outliers	Valeurs des outliers	Proportion des outliers (en %)
ï..Age	non	aucun	0.00
sex	non	aucun	0.00
chestPainType	non	aucun	0.00
restingBloodPressure	oui	172 180 180 178 192 200 180 178 174	3.33
serumCholestoral	oui	394 409 417 407 564	1.85
fastingBloodSugar	non	aucun	0.00
restingElectrocardiographicResults	non	aucun	0.00
maximumHeartRateAchieved	oui	71	0.37
exerciseInducedAngina	non	aucun	0.00
oldpeak	oui	6.2 4.2 5.6 4.2	1.48
ST	non	aucun	0.00
vessels	non	aucun	0.00
thal	non	aucun	0.00
HeartDisease	non	aucun	0.00

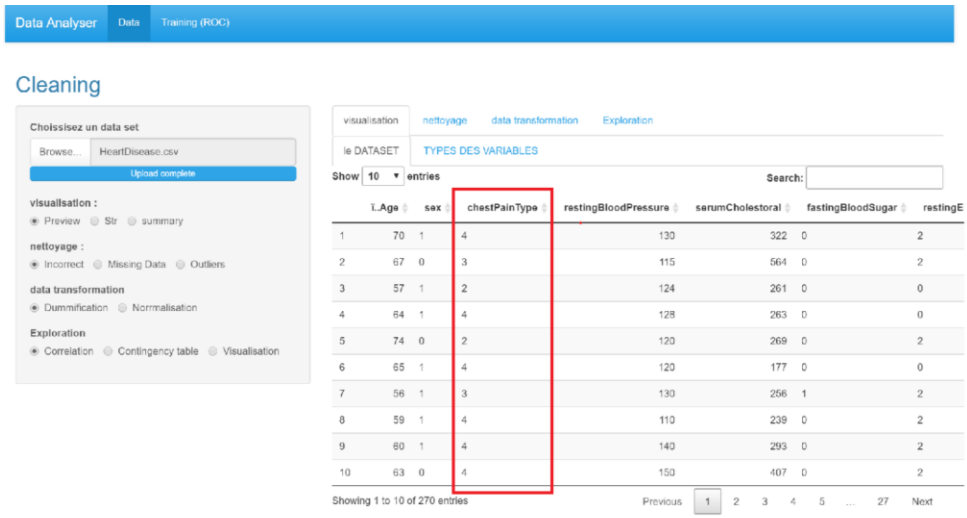
En rouge nous avons la suppression des lignes contenant des outliers et en vert le remplacement des lignes contenant des outliers par la moyenne par exemple. Dans les select il y a les variables présentant des outliers.

2.3.Data transformation

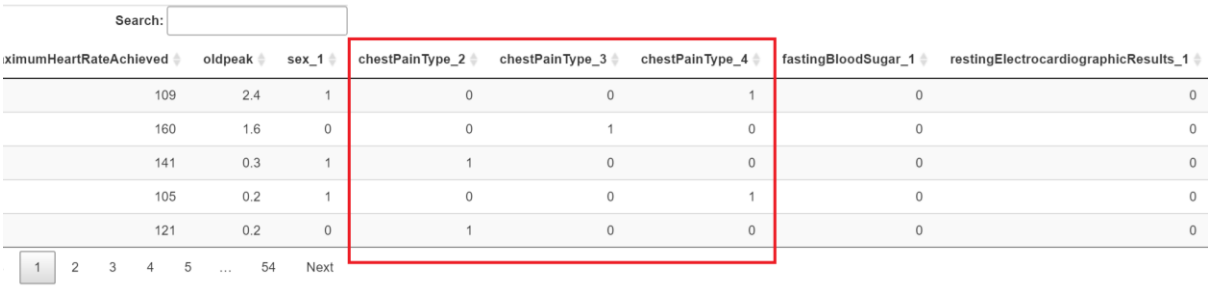
2.3.1.Dummification

Il s'agit de créer pour chaque variable de type factor K-1 nouvelles variables binaires, voici un exemple de dummification avec la variable **chestPainType**:

Avant dummification:

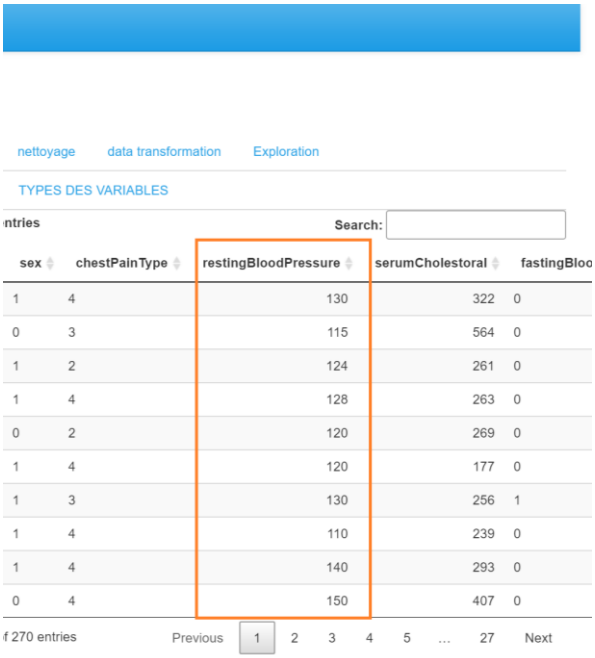


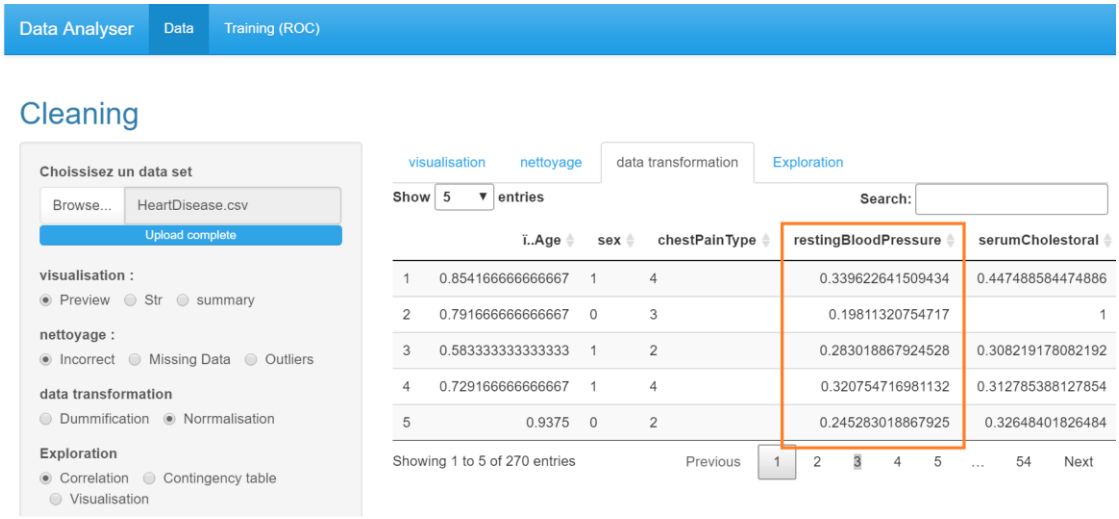
Après dummification:



2.3.2.Normalisation

- La normalisation consiste à modifier la valeurs des données de telle sorte à obtenir des valeurs comprises entre 0 et 1. Elle s'applique aux variables quantitatives continues, par exemple pour la variable **restingBloodPressure** on aura :





2.4.Exploration des données

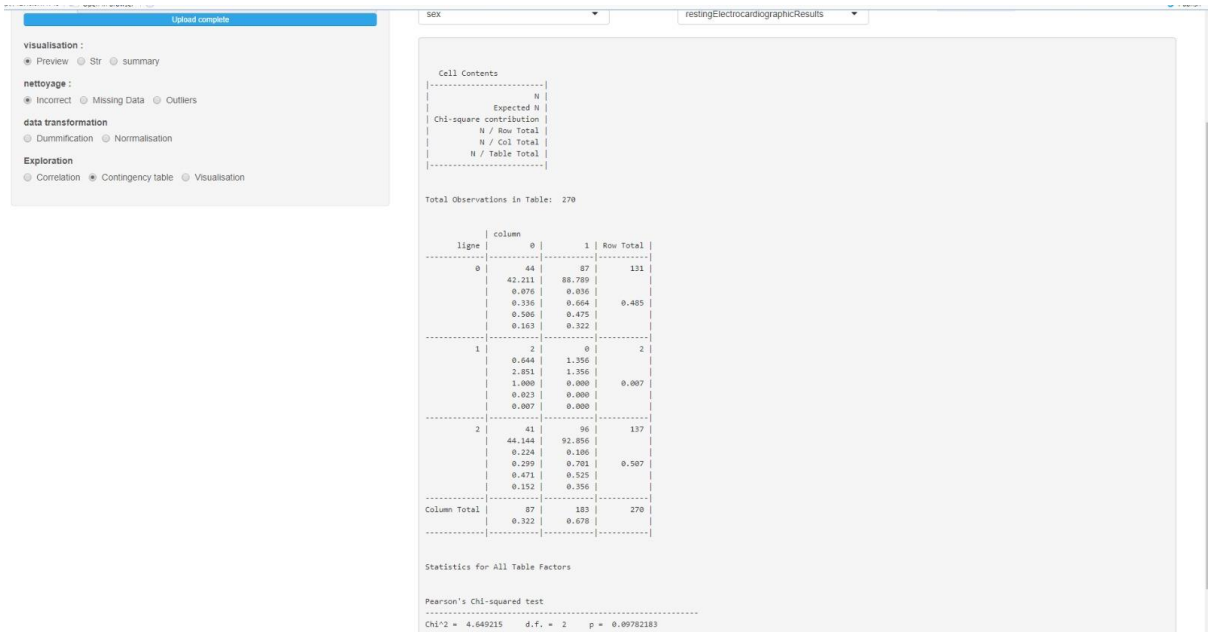
2.4.1.Correlation

Notre dataset contient quatre variables continues: Âge, Oldpeak, serumCholesterol et MaximumHearRateAchieved. A l'aide de la matrice de corrélation présentée dans la capture d'écran ci-dessous on note l'existence d'une forte corrélation entre MaximumHearRateAchieved et Age ainsi

que MaximumHearRateAchieved et oldpeak.

2.4.2.Table de contingence

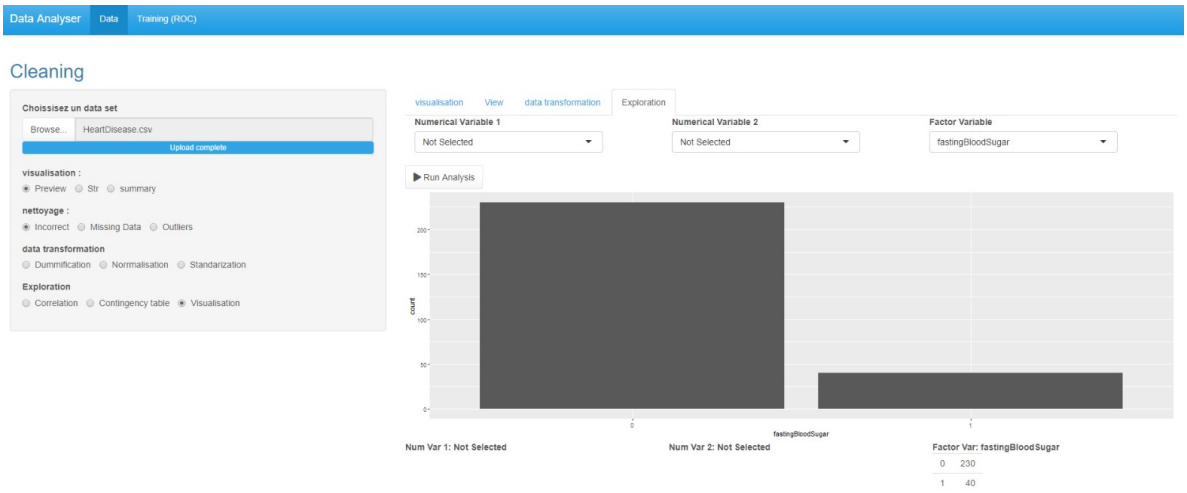
En choisissant deux variable qualitative nous obtenons:



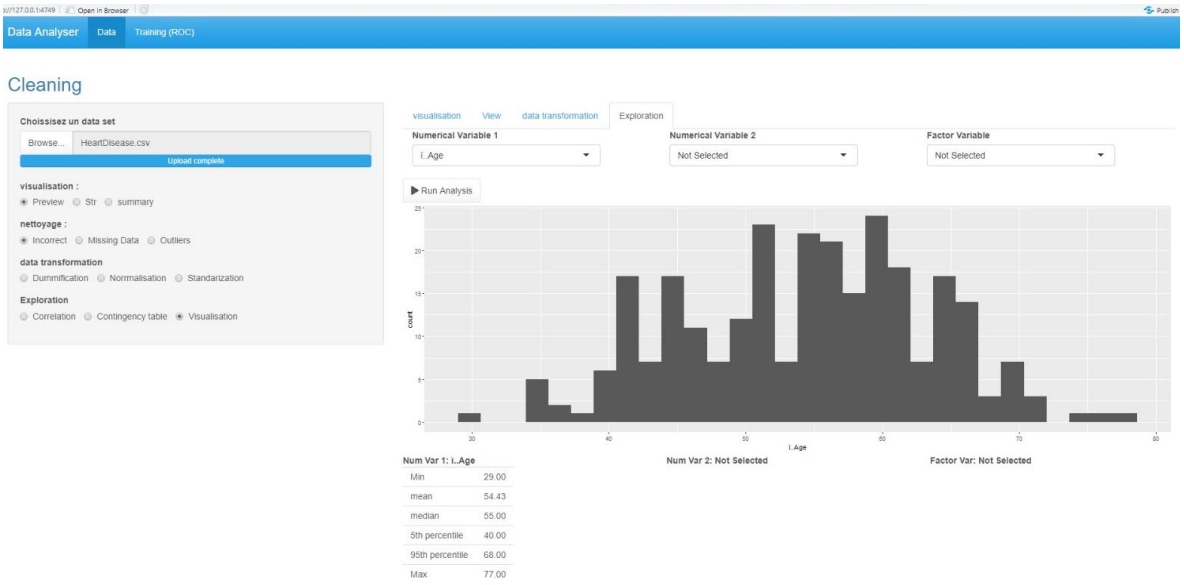
2.4.3.Visualisation

Dans la partie visualisation, l'utilisateur peut sélectionner jusqu'à 2 variables numériques et 1 variable de type facteur (les deux premiers boutons select permettent de sélectionner une variable quantitative (numérique) et le 3ème bouton select permet de sélectionner une variable qualitative (factor)). Ainsi, lorsque l'on sélectionne uniquement une variable quantitative cela nous affiche son diagramme en baton de même lorsque l'on sélectionne uniquement une variable qualitative (3ème bouton select). Lorsque l'on sélectionne deux variables quantitatives cela nous affiche un nuage de point et lorsque l'on ajoute en plus une variable qualitative cela nous affiche un nuage de point en fonction des 3 variables sélectionnées (A noter que les deux premiers boutons selects n'affiche que les variables quantitative (type numérique) et le troisième bouton select n'affiche que les variables qualitative (type qualitative)). Voici une illustration de l'utilisation de cette partie:

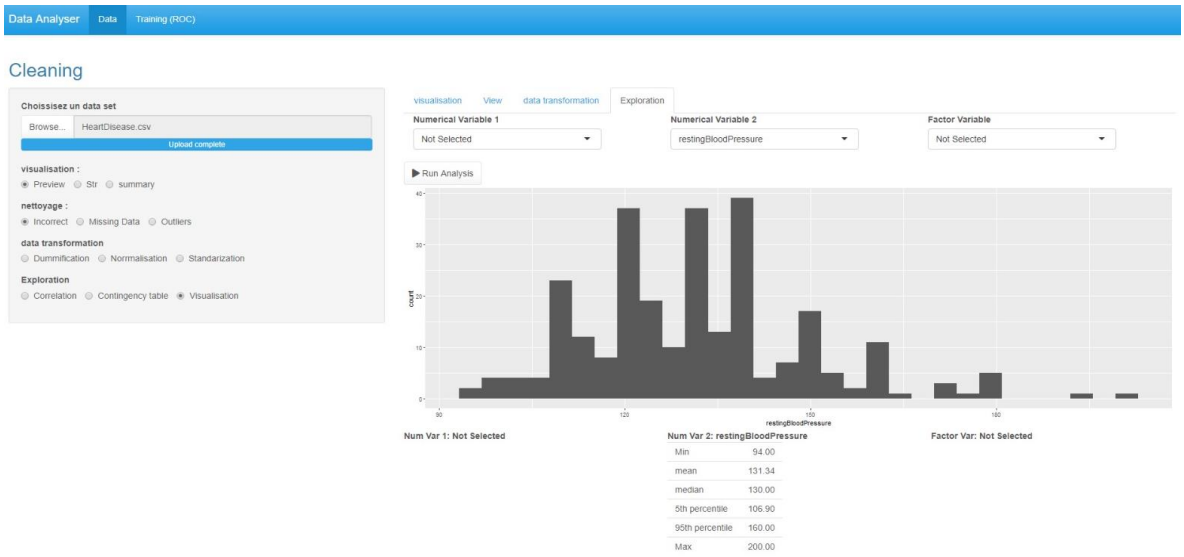
- Sélection d’une variable qualitative seulement (au niveau du troisième bouton select):



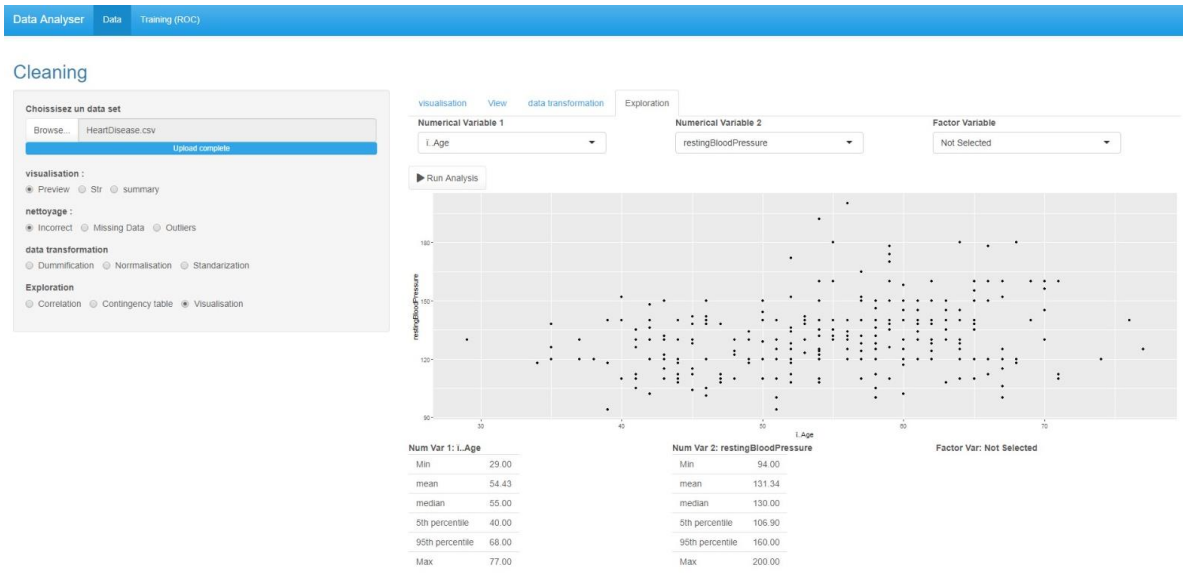
- Sélection d’une variable quantitative seulement (au niveau du premier bouton select):



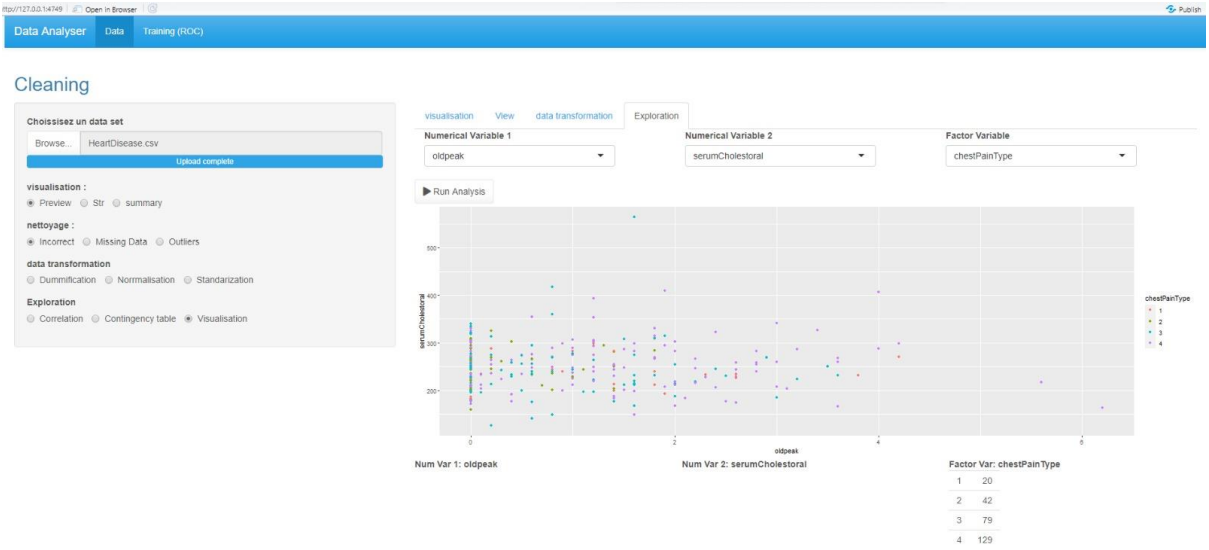
- Sélection d’une variable quantitative seulement (2ème bouton select):



- Sélection de deux variables quantitative (1ème et 2ème boutons select):

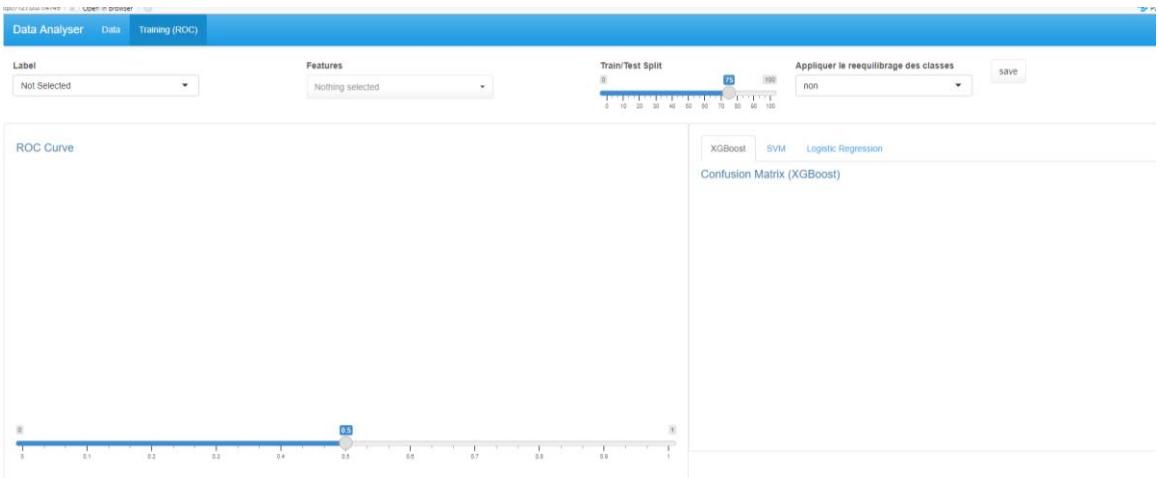


- Sélection de 2 variables quantitatives et 1 variable qualitative (1er, 2ème et 3ème bouton select respectivement):



2.5.Entraînement des modèles

L'onglet training nous permet de visualiser trois modèles statistique préalablement choisis: Xgboost, Régression Logistique et SVM (Support Vector Machine). Elle permet de choisir pour le bouton select "Label" la variable cible sur laquelle on souhaite effectuer nos prédictions. Ce bouton select n'affiche que les variable de type factor à deux levels. Le deuxième bouton select "Features" permet de choisir différentes variables explicatives. Cette liste déroulante se met à jour lorsque l'on choisit une variable cible et permet ainsi d'afficher dans la liste "Features" toutes les variables mise à part le Label choisi. Cela est possible grâce aux observerEvent. Il y a également un slider permettant de choisir la division train et test. Cette partie permet aussi d'afficher l'équilibrage des différentes classes de la variable cible choisie pour le train et de les rééquilibrer si l'utilisateur en sent le besoin. Cet onglet permet d'afficher les courbes ROC de chaque modèle mais également d'afficher la matrice de confusion et les metric AUC, accuracy, F-score et recall.



3. Choix du Dataset et analyse des résultats obtenus:

Notre choix s'est porté sur le dataset **HeartDiseases** qui est une base de données sur les maladies cardiaques. Il contient 13 attributs et 270 observations. La variable à prédire est **heartDisease** qui indique si oui ou non le patient a une maladie cardiaque.

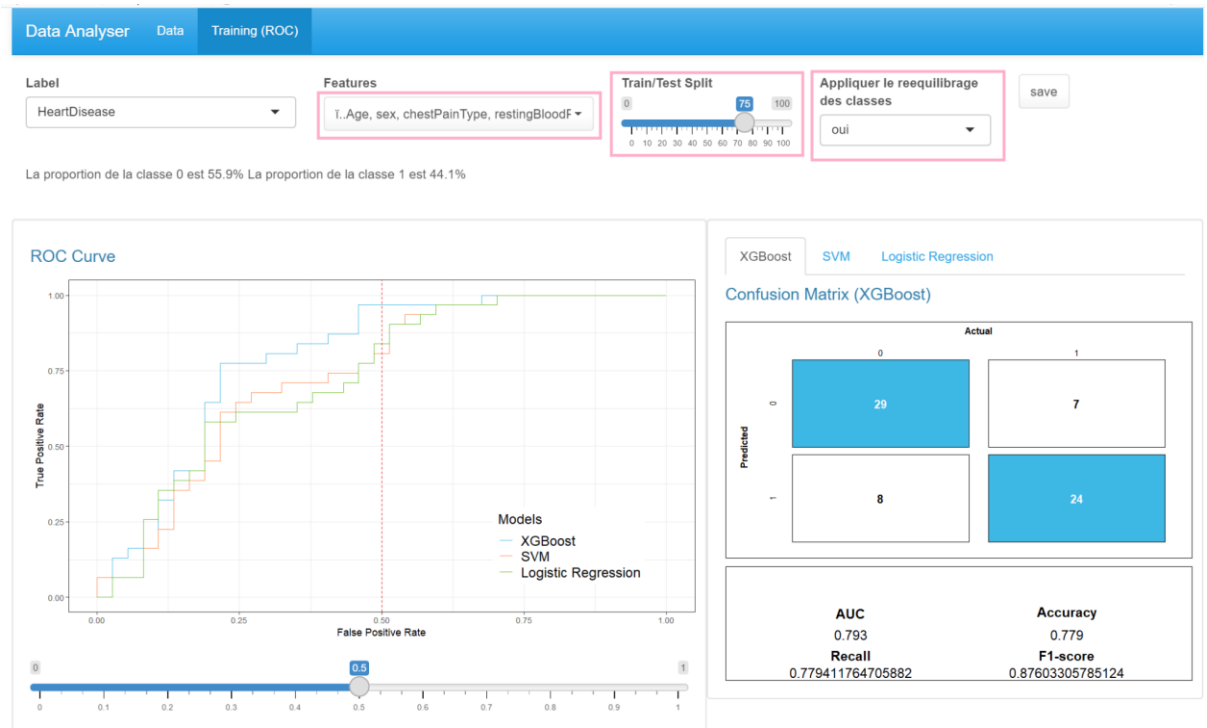
Dans ce qui suit nous allons interpréter les résultats obtenus suite à la phase d'apprentissage de nos différents modèles pour ensuite décider quelle méthode d'apprentissage est la plus adaptée à ce problème de classification.

L'évaluation de la performance des modèles a été faite sur la base de la matrice de confusion ainsi que les métriques AUC, Accuracy, Recall et F-score

3.1.XGboost

Dans un premier temps on choisit les attributs age, chestPainType et restingBloodPresure. On fait un découpage de 75% pour la partie training et 25% pour la partie test puis on applique un équilibrage de classes ce qui nous permet d'avoir 55,9% des instances de la classe 0 et 44,1% des instances de la classe 1.

On obtient les résultats suivant :

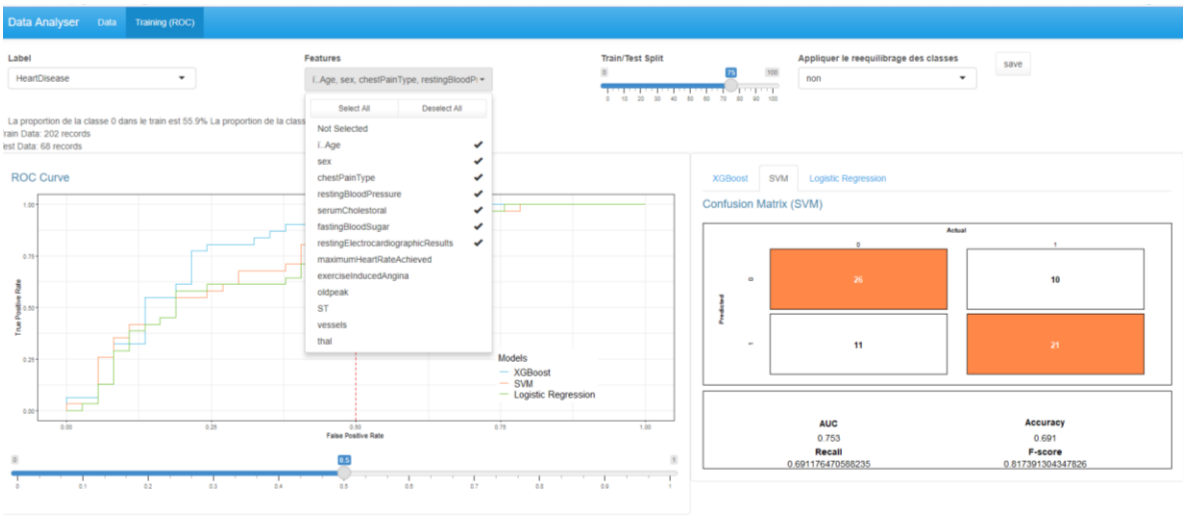


On choisissant les attributs age, sex,chestPainType et serumCholes et en faisant un découpage de 82% pour la partie training et 18% pour la partie test puis en appliquant un équilibrage de classes ce qui nous permet d'avoir 55,7% des instances de la classe 0 et 44,3% des instances de la classe 1, on obtient de meilleurs résultats

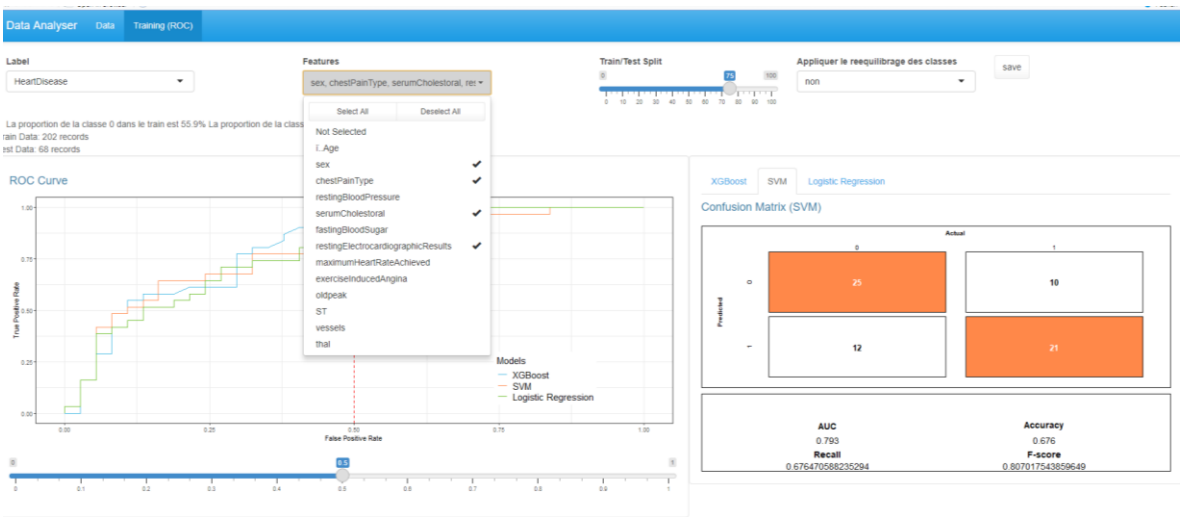
Après plusieurs essais on peut conclure que les variables pertinentes sont Age, sex, chestPainType et serumCholes car elles nous ont permis d'avoir les meilleures prédictions selon les metrics d'évaluation des modèles calculées .

3.2.SVM

Pour le modèle SVM (Support Vector Machine) nous avons choisi dans un premier temps les variables Age, sex, chestPainType, restingBloodPressure, serumCholestoral, fasting BloodSugar et restingElectrocardiographicResults. Nous effectuons également un découpage de 75% pour la partie training et 25% pour la partie test et obtenons ceci:

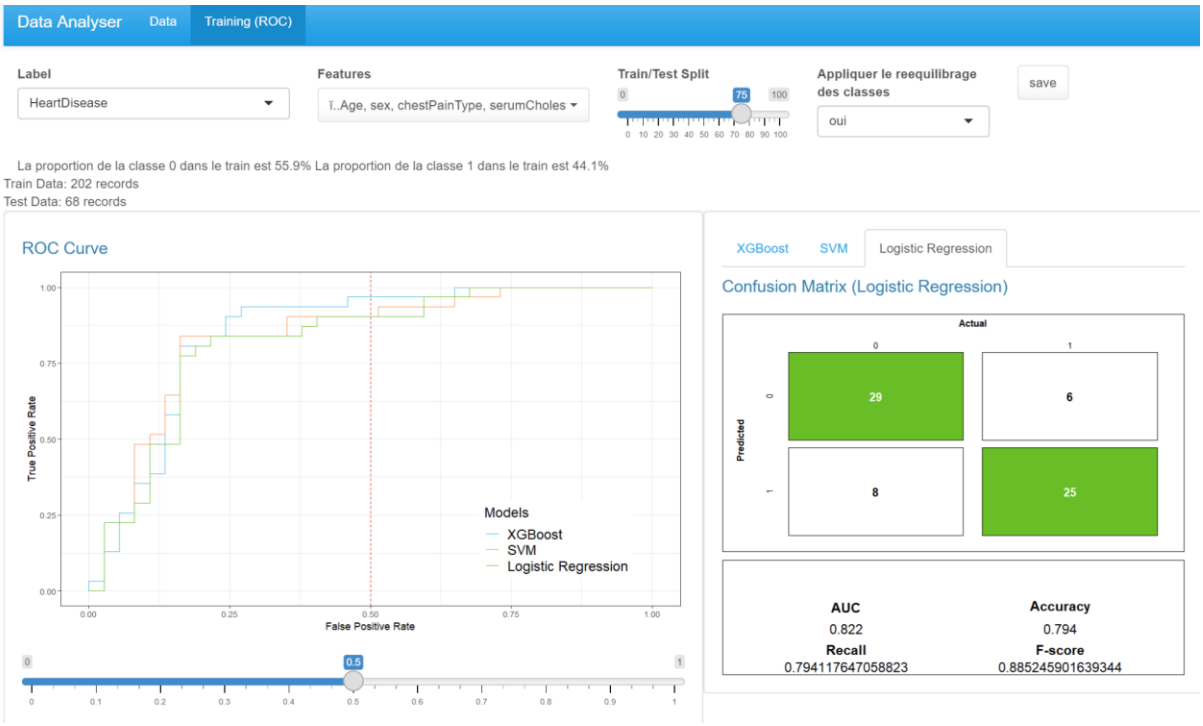


Puis au fur et à mesure, en fonction de l'AUC nous avons enlevé différentes variables jusqu'à trouver la combinaison de variables qui permet d'avoir la meilleure valeur d'AUC représentant ainsi les variables pertinentes. Ces variables sont sex, chestPainType, serumCholestoral et restingElectrocardiographicResults.

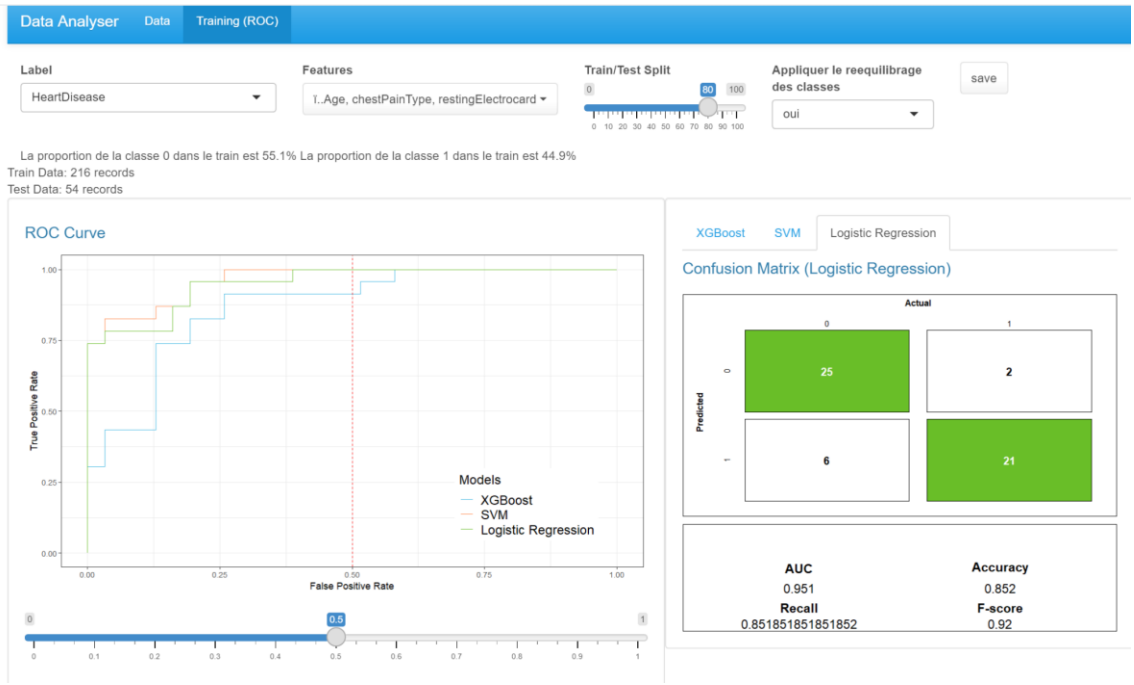


3.3.Régression Logistique

Dans un premier test, on choisit l'attribut age, sex, chestPainType et serumCholes, et on entraî
notre modèle sur 75% de notre dataset original et on réalise l'équilibrage de classes, les résultats sont
suivants:



On choisit l'attribut age, chestPainType et restingElectrocrd, et on entraîne notre modèle sur 80% de notre dataset original et on réalise l'équilibrage de classes, les résultats sont les suivants:



Il est évident d'après les résultats obtenus que la variable âge combinée à chestPainType et restiongElectrocord donne de meilleurs résultats, on pourra ainsi dire que ces dernières représentent les variables les plus pertinentes.

Conclusion:

Notre application nous permet d'explorer n'importe quel jeux de données et plus particulièrement le dataset HeartDisease que l'on a choisi. Cette interface Web permet également comme précisé auparavant d'entraîner des modèles de classification et ainsi pouvoir prédire les valeurs d'une variable cible.

Etant donné que notre dataset contient 13 attributs, il sera plus judicieux de choisir quelques variables pour apprendre notre modèle (les variables les plus pertinentes et qui influencent le plus la variable cible) car si on choisit beaucoup de variables on pourra avoir un overfitting. Donc plusieurs tests ont été effectués avec différentes combinaisons de variables. A chaque test, on prenait en compte les valeurs des métriques obtenues pour au final opter pour la combinaison qui a donné les valeurs des métriques les plus élevées et qui est constituée des variables âge, chestPainType et restiongElectrocord.

Ainsi, après avoir testé différentes méthodes d'apprentissage pour la résolution du problème de classification qui consiste à prédire si oui ou non un malade a des problèmes cardiaques, nous pouvons conclure que le modèle de la régression logistique est le plus adapté car il nous a donné des valeurs de métriques élevées par rapport à celles des autres modèles.