Article sur la prédiction d'avc

February 23, 2022

1 DEMARCHE DE TRAVAIL

1.1 OBJECTIF MESURABLE

OBJECTIF : Prédire si une personne est atteinte d'un accident vasculaire cérébral ou pas à partir des données personnelles et cliniques fournies (problème de classification). Il est essentiel de noter qu'il est plus urgent, dans ce cas de figure, de détecter toutes les personnes malades (et non celles qui sont vraiment malades parmi ceux qui sont identifiés comme malade). Sinon un patient malade peut être diagnostiqué comme sain alors que ce n'est pas le cas. Nous préférerions ainsi la sensibilité à la précision.

METRIQUE ET SCORE A ATTEINDRE : La proportion de la classe positive est largement inférieure à la proportion de la classe négative (nous notons une forte déséquilibre de classes). Dans ce cas de figure, la métrique accuracy ne sera pas assez performante pour évaluer notre modèle de classification. A la place, nous allons utiliser les métriques sensibilité et précision pour valider notre modèle :

- **Sensibilité** : $\frac{VP}{VP+FN}$. Elle permet de calculer le pourcentage de tests positifs parmi les patients réellement atteints d'AVC ;
- **Spécificité** : $\frac{VN}{VN+FP}$. Elle permet de calculer le pourcentage de tests négatifs parmi les patients réellement sains. C'est la sensibilité pour les test négatifs ;
- **Précision** : $\frac{VP}{VP+FP}$. Elle permet de calculer le pourcentage de patients réellement malades parmi ceux dont le test est positif ;
- **F1-score** : $\frac{2\times Sensibilite\times Precision}{Sensibilite+Precision}$;
- Avec :
  - VP : Nombre de Vrais Positifs ;
  - VN : Nombre de Vrais Négatifs ;
  - FN : Nombre de Faux Négatifs ;
  - FP : Nombre de Faux Positifs ;

Nous nous fixons une sensibilité supérieure à **80%**. Nous pouvons également utiliser des outils supplémentaires comme **les courbes ROC et Precision-Recall** pour affiner nos choix de performance. Nous pouvons nous fixer une aire sous la courbe (AUC) supérieure ou égale à **80%**.

1.2 EXPLORATION DES DONNEES

1.2.1 ANALYSE DE LA FORME

CIBLE : La variable dépendante est la variable **stroke** qui contient des données de type discrète. Cette variable indique si une personne est atteinte d'un accident vasculaire cérébral (1 pour atteinte) ou pas (0 pour non atteinte). Nous allons abréger, par la suite, le terme *accident vasculaire cérébral* par *AVC* pour faciliter l'écrit. Nous devons transformer, plus tard, la variable cible en variable catégorielle non ordinale (cela nous permettra de différencier la catégorie "être atteint(e) d'AVC" de la catégorie "être non atteint(e) d'AVC"). Nous devrons ainsi utiliser un modèle de classification (classification model) pour déterminer si un patient est malade ou sain.

NOMBRE DE LIGNES ET DE COLONNES : Nous avons identifié 5110 observations et 12 variables (dont la cible). Le nombre d'observations est inférieur à 10000, donc notre dataset ne contient pas énormément d'observations mais assez pour entraîner un modèle.

TYPES DE VARIABLES : Parmi les variables explicatives, nous avons identifié 7 variables catégorielles (dont deux contiennent des valeurs discrètes et les autres contiennent des valeurs de type chaîne de caractères) et 3 variables non catégorielles.

- Les variables de type object ont pour valeurs possibles les suivantes :
  - **gender** (genre) : contient les valeurs *Male* (Homme), *Female* (Femelle) ou *Other* (ni Homme, ni Femme) ;
  - **ever_married** (jamais marié(e)) : peut prendre les valeurs *Yes* (Oui) ou *No* (Non) ;
  - **work_type** (type de travail) : peut prendre les valeurs *Private* (Privée), *Self_employed* (Auto emploi ou Travail autonome), *Govt_job* (Travail au gouvernement), *children* (enfant), *Never_worked* (N'a jamais travaillé(e)) ;
  - **Residence_type** (type de résidence) : peut prendre les valeurs *Urban* (Urbaine), *Rural* (Rurale) ;
  - **smoking_status** (statut de fumeur) : contient les valeurs *formerly_smoked* (a fumé(e) dans le passé), *never_smoke* (n'a jamais fumé), *smokes* (fume), *Unknow* (donnée non recueillie).
- Quant aux variables catégorielles **hypertension** et **heart_disease** (maladie de coeur), elles peuvent prendre les valeurs 1 (pour *positive*) ou 0 (pour *négative*).
- Nous identifions une variable discrète non categorielle (**age**) dont les valeurs varient de 0.08 (enfant de 9 mois) à 82 (adulte de 82 ans). Sa valeur moyenne est de 43 (adulte de 43 ans).
- Le dataset ne contient que deux variables continues, **avg_glucose_level** (niveau moyen de glucose dans le sang) et **bmi** (indice de masse corporelle). Ces deux variables ne s'expriment pas avec les mêmes unités :
  - La variable avg_glucose_level a pour valeur maximale 271.74 et pour valeur minimale 55.12. Sa valeur moyenne est de 106.15.
  - La variable bmi a pour valeur maximale 97.60 et pour valeur minimale 10.30. Sa valeur moyenne est 28.89.
- Nous définirons plus en détail ces variables dans la partie analyse du fond.

IDENTIFICATION DES VALEURS MANQUANTES : Nous remarquons que seule la variable bmi contient des valeurs manquantes qui sont un peu dispersées dans le dataset. Mais en

sachant que les valeurs manquantes ne représentent que 16% de l'ensemble des données présentes dans la colonne bmi et que nous allons choisir un échantillon aléatoire pour l'entraînement du modèle (ainsi que pour l'évaluation) donc on peut supposer que le remplacement des valeurs manquantes par la valeur la plus fréquente de la variable bmi constitue une bonne stratégie.

IDENTIFICATION DES VALEURS REDONDANTES : Les données ne contiennent pas d'observations dupliquées.

1.2.2 ANALYSE DU FOND

VISUALISATION DE LA CIBLE : Nous remarquons que 95.13% des patients sont atteintes d'AVC contre seulement 4.87% des patients atteintes d'AVC. La proportion de patients non malades est largement supérieure au nombre de patients non malades.

COMPREHENSION DES DIFFERENTES VARIABLES :

- **id** : La variable id contient des valeurs discrètes et identifie chaque observation de manière unique. Elle n'apporte aucune information supplémentaire et donc n'influence pas le fait qu'une personne soit malade ou non. La variable id doit être supprimée.
- Variables catégorielles :
  - **Hypertension** : Cette variable indique si oui ou non, le patient souffre d'hypertension. Un patient est testé positif à l'hypertension si on constate à deux reprises, et pas dans le même jour, une tension systolique supérieure ou égale à 140 mm Hg et/ou une tension diastolique supérieure ou égale à 90 mm Hg. Selon les tests cliniques, une hypertension peut souvent être la cause d'AVC. Nous verrons par la suite si les données recueillies de cette variable sont fiables. La colonne hypertension contient 90% de tests négatifs et 10% de tests positifs.
  - **heart_disease** : La variable heart_disease indique si le patient est atteint de cardiopathie (maladie cardiaque) ou pas. Il y a différents types de cardiopathies et nous verrons si ces derniers peuvent influencer le risque d'attraper un AVC. La colonne heart_disease contient 95% de tests positifs contre seulement 5% de tests négatifs. Cependant, nous savons que l'hypertension non traitée peut causer la cardiopathie. Donc on peut supposer, d'ores et déjà, que ces deux variables sont fortement corrélées (hypothèse à vérifier).
  - **gender** : La variable gender indique à quel sexe appartient le patient. On doit vérifier si le sexe du patient peut influencer le risque qu'il soit atteint d'AVC. La colonne gender est composée de 58% de femmes, de 41% d'hommes et très peu (presque 0%) de sexe de type autre.
  - **ever_married** : Cette variable indique si le patient a déjà été marié. L'analyse de cette variable doit nous permettre de dire si le patient a plus de chance d'attraper un AVC en s'étant déjà marié (dans le temps présent ou passé) ou pas. Nous notons 66% des patients qui se sont jamais mariés et 34% qui se sont déjà mariés.
  - **work_type** : Elle indique le type de travail effectué par un patient. On identifie 57% de patients travaillant dans le secteur privé (professions et secteurs d'activité ne dépendant pas de l'Etat), 16% des patients effectuant du travail autonome (ils sont leurs propres employés), 13% des patients sont des enfants (on n'a pas plus d'informations sur la catégorie children mais on suppose pour l'instant que le patient est un enfant et donc qu'il n'a pas besoin de travailler), presque 13% des patients travaillent pour le

gouvernement (dans le secteur public) et seulement 0.4% n'ont jamais travaillé. Pour les patients dont le type de travail est children, nous devons vérifier si leurs âges indiquent que ce sont des enfants ou pas (c'est-à-dire certains sont des adultes).
  - **residence_type** : Cette variable indique le type de résidence du patient. 51% des patients résident dans un milieu urbain (ville) et 49% des patients résident dans un milieu rural (campagne). Les proportions de ces deux classes sont presque similaires.
  - **smoking_status** : Elle indique si le patient est/était un fumeur ou pas. 37% des patients n'ont jamais fumé, 30% des patients n'indiquent pas s'ils fument, 17% des patients ont fumé auparavant et 15% des patients disent fumer au moment où on les interrogeait. La catégorie unknow (inconnue) peut constituer un problème car elle n'apporte aucune information utile. Nous vérifierons par la suite si cette variable apporte de l'information à notre modèle.

- Variables non catégorielles :

  - **age** : Nous constatons que les patients qui sont âgés entre 37 et 63 ans sont plus nombreux dans la base de données, suivis des patients qui sont âgés entre 78 et 82 ans tandis que les plus jeunes sont les moins nombreux.
  - **avg_glucose_level** : Le niveau moyen de glucose dans le sang indique si, oui ou non, le patient est atteint de diabète. Cette variable s'exprime en mg/dL (milligrammes par décilitre). Les patients qui souffrent de diabète ont un niveau moyen de glucose supérieur ou égale à 200 mg/dL. Par contre, un niveau moyen de glucose, inférieur à 140 mg/dL, est considéré comme normal. Nous constatons que la plupart des patients n'ont pas de diabète car une grande partie des patients ont un niveau moyen de glucose tournant autour de 75-87 mg/dL de sang. Nous remarquons qu'environ 10 à 20% des patients ont un niveau moyen de glucose tournant autour de 210-225 mg/dL. On peut considérer que ces derniers sont atteints de diabète.
  - **bmi** : L'indice de masse corporelle (IMC) permet d'indiquer la corpulence d'un patient. Il s'exprime en kg/m2 (kilogrammes par mètre carré). Une personne est considérée comme obèse si son IMC dépasse 30 kg/m2. La plupart des patients examinés ont un IMC situé autour de 29 kg/m2. Ces derniers présentent un cas de surpoids ou d'obésité modérée (dont les IMC sont respectivement situés entre 25 et 30 kg/m2 et entre 30 et 35 kg/m2).

VISUALISATION DES RELATIONS ENTRE LES VARIABLES EXPLICATIVES ET LA CIBLE :

- Relations Cible - variables catégorielles :

  - Pour la variable work_type : Nous remarquons que les patients qui ont pour type de travail children ont plus de chance de ne pas attraper d'AVC que les patients qui effectuent d'autres types de travail. On a peu de patients non travailleurs dans la variable work_type donc on ne peut rien dire par rapport à cette catégorie (mais il est possible qu'elle influence aussi le fait qu'on soit atteint d'AVC ou pas).
  - Pour la variable smoking_type : Nous remarquons que les patients ne disant pas s'ils fument/fumaient présentent une proportion de malades inconsistante par rapport aux autres types de fumeurs. Cela est dû au fait que la catégorie unknow n'est pas fiable (elle se comporte comme une valeur manquante).

- Relations Cible - Variables non catégorielles :

- Parmi les variables non catégorielles, seule la variable âge influence le fait qu'un patient soit atteint d'AVC ou pas. Il y a plus de risque d'attraper un AVC chez les patients plus âgés ;
- Pour les deux autres variables restantes nous constatons que les distributions sont presque les mêmes pour les patients malades et non malades ;
- Nous vérifierons plus amplement ces hypothèses à travers un test de student.

VISUALISATION DES RELATIONS ENTRE VARIABLES QUANTITATIVES :

Les variables quantitatives ne partagent aucune forte corrélation entre elles.

VISUALISATION DES RELATIONS ENTRE VARIABLES CATEGORIELLES ET QUANTITATIVES :

Nous constatons que quelques variables catégorielles ont de fortes corrélations avec des variables quantitatives.

- La variable **age** est fortement corrélée avec les variables **work_type**, **smoking_status** et **ever_married**. Pour sa relation avec les autres variables catégorielles nous remarquons de légères corrélations ou quasiment pas de corrélations pour certaines.
- La variable **bmi** est, pour son cas, fortement corrélée avec les variables **ever_married** et **work_type**.
- En revenant au niveau de pandas profiling, nous constatons que nos analyses sont soutenues par les remarques faites par la librairie (aucune incohérence n'est à noter).

IDENTIFICATION DES VALEURS ABERRANTES :

Nous constatons que quelques valeurs dépassent la limite et peuvent être considérées comme aberrantes pour le moment.

- La variable age ne comporte presque pas de valeurs aberrantes.
- Les variables bmi et avg_glucose_level contiennent par contre des valeurs aberrantes.
- La variable avg_glucose_level ne contient des valeurs aberrantes que du côté de la classe *test negatif* de la variable stroke alors que la variable bmi en comporte pour les deux classes (mais beaucoup plus du côté de la classe *test negatif*).
- Cependant pour la variable avg_glucose_level nous constatons que ces valeurs coïncident avec les valeurs désignant le groupe de patients atteints de diabète. Donc nous devrons traiter ces valeurs avec précaution car elles peuvent être utiles à l'identification de patients atteints d'AVC.

PREMIERE CONCLUSION :

- On note une forte désiquilibre de classes au niveau de la cible. Le pourcentage de patients non atteints dépasse les **90%**. Ce problème peut-être traité avec l'équilibrage des classes à l'aide de quelques techniques de rééchantillonnage (sous échantillonnage, sur échantillonnage ou les deux).
- On constate une dépendance entre la variable cible et la variable age.
- La variable id ne comporte aucune information utile donc il est préférable de la supprimer de la base de données.
- La base de données contient des valeurs manquantes que nous allons remplacer par la valeur la plus fréquente de la variable bmi ;
- La catégorie unknow de la variable smoking_status n'apporte aucune information supplémentaire donc nous devons la remplacer par une catégorie plus intéressante (*never smoked*

par exemple) ;
- Les variables quantitatives ne sont pas corrélées entre elles ;
- La variable age est fortement corrélée avec certaines variables catégorielles. Notamment les variables work_type, ever_married et smoking_status ;
- La variable bmi est aussi fortement corrélée avec les variables catégorielles ever_married et work_type ;
- Nous devons garder les variables age et bmi et supprimer les variables smoking_status, work_type et ever_married ;
- Des tests supplémentaires seront réalisé pour étayer certaines hypothèses ou les rejeter ;
- Les variables quantitatives comportent des valeurs 'anormales'. Nous allons vérifier s'il est nécessaire de les remplacer ou de les supprimer. Ces données peuvent être utiles car ils caractérisent un groupe de patients.
- Nous allons retenir pour l'instant les variables hypertension et heart_disease qui sont cliniquement très intéressantes pour nos analyses.

TESTS D'HYPOTHESES : Ces tests nous permettrons de valider ou de rejeter certaines hypothèses.

- Testons si les variables quantitatives influencent le fait qu'un patient soit atteint d'AVC ou pas (nous noterons H0 l'hypothèse selon laquelle une variable n'a pas d'influence sur la cible) :
  - Le test de student nous indique que les variables **age** et **avg_glucose_level influencent le fait qu'un patient soit atteint d'AVC ou pas**.
  - La variable **bmi n'entretient pas de dépendance avec la variable stroke**.

SECONDE CONCLUSION :

- Les variables **age** et **avg_glucose_level** apportent de l'information contrairement à la variable bmi qui, elle, peut-être supprimée.
- Les variables à supprimer sont donc : **bmi**, **work_type**, **ever_married**, **smoking_status**, **id**. Pour le moment nous ne supprimerons que ces variables. Nous ferons une sélection antérieure de variables si on constate un sur entraînement du modèle.

2 Exploration des données

2.1 Identification de la cible

Contenu des dix premières lignes.

	id	gender	age	hypertension	heart_disease	ever_married
0	9046	Male	67.0	0	1	Yes
1	51676	Female	61.0	0	0	Yes
2	31112	Male	80.0	0	1	Yes
3	60182	Female	49.0	0	0	Yes
4	1665	Female	79.0	1	0	Yes
5	56669	Male	81.0	0	0	Yes
6	53882	Male	74.0	1	1	Yes
7	10434	Female	69.0	0	0	No

```
8  27419  Female  59.0            0              0          Yes
9  60491  Female  78.0            0              0          Yes

       work_type Residence_type  avg_glucose_level   bmi   smoking_status  \
0        Private          Urban             228.69  36.6  formerly smoked
1  Self-employed          Rural             202.21   NaN     never smoked
2        Private          Rural             105.92  32.5     never smoked
3        Private          Urban             171.23  34.4           smokes
4  Self-employed          Rural             174.12  24.0     never smoked
5        Private          Urban             186.21  29.0  formerly smoked
6        Private          Rural              70.09  27.4     never smoked
7        Private          Urban              94.39  22.8     never smoked
8        Private          Rural              76.15   NaN          Unknown
9        Private          Urban              58.57  24.2          Unknown

   stroke
0       1
1       1
2       1
3       1
4       1
5       1
6       1
7       1
8       1
9       1
```

Nous savons que la cible est la variable stroke. C'est la variable à expliquer.

Type de la cible

```
La cible a pour type :  int64
```

Valeurs possibles de la cible

```
La cible est composée des modalités :  [1, 0]
```

2.2 Nombre de ligne et de colonnes

```
Nous avons au total 5110  lignes et  12  colonnes dans le jeu de données.
```

2.3 Nombre de ligne et de colonnes

Types des colonnes.

```
id                   int64
gender              object
age                float64
hypertension         int64
heart_disease        int64
```

```
ever_married          object
work_type             object
Residence_type        object
avg_glucose_level    float64
bmi                  float64
smoking_status        object
stroke                 int64
dtype: object
```

Nous identifions 7 colonnes de type catégorielles.

Les modalités possibles des variables de type *chaîne de caractères*.

```
Valeurs uniques de gender-------------- ['Male' 'Female' 'Other']

Valeurs uniques de ever_married-------- ['Yes' 'No']

Valeurs uniques de work_type----------- ['Private' 'Self-employed' 'Govt_job'
'children' 'Never_worked']

Valeurs uniques de Residence_type------ ['Urban' 'Rural']

Valeurs uniques de smoking_status------ ['formerly smoked' 'never smoked'
'smokes' 'Unknown']
```

Les modalités possibles des variables *hypertension* et *heart_disease*.

```
Modalités de la colonne hypertension------------------ [0 1]

Modalités de la colonne heart_disease----------------- [1 0]
```

Description (résumé) des données quantitatives.

	age	avg_glucose_level	bmi
count	5110.000000	5110.000000	4909.000000
mean	43.226614	106.147677	28.893237
std	22.612647	45.283560	7.854067
min	0.080000	55.120000	10.300000
25%	25.000000	77.245000	23.500000
50%	45.000000	91.885000	28.100000
75%	61.000000	114.090000	33.100000
max	82.000000	271.740000	97.600000

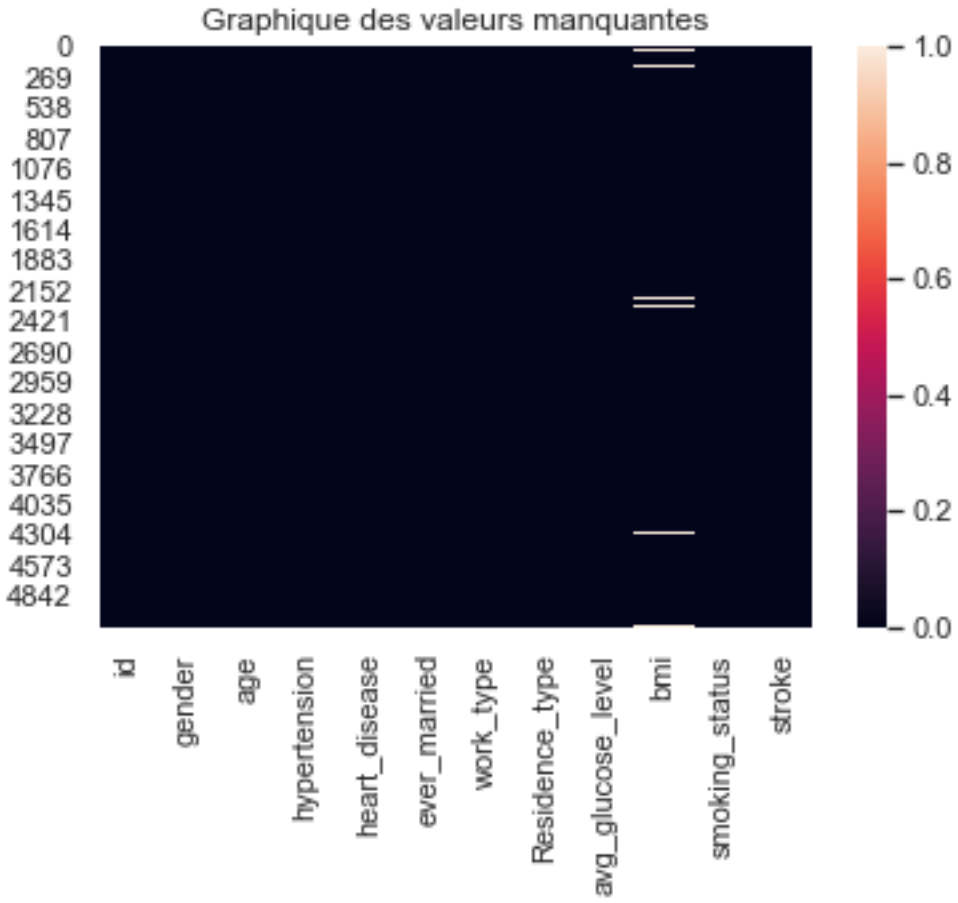
2.4 Identification des valeurs manquantes

Première analyse des valeurs manquantes par sommation.

```
id                   0
gender               0
age                  0
hypertension         0
heart_disease        0
ever_married         0
work_type            0
Residence_type       0
avg_glucose_level    0
bmi                201
smoking_status       0
stroke               0
dtype: int64
```

Deuxième analyse par visualisation (Visualisation de la répartition des valeurs manquantes dans le dataset).

```
Text(0.5, 1.0, 'Graphique des valeurs manquantes')
```



Nous remarquons que seule la variable bmi contient des valeurs manquantes.

pourcentage de valeurs manquantes dans la variable bmi.

```
La variable bmi contient 16.75% de valeurs manquantes.
```

2.5 Identification des observations redondantes

```
Le nombre d'observations redondantes est de  0
```

Le dataset ne contient pas d'observations dupliquées.

2.6 Visualisation de la cible

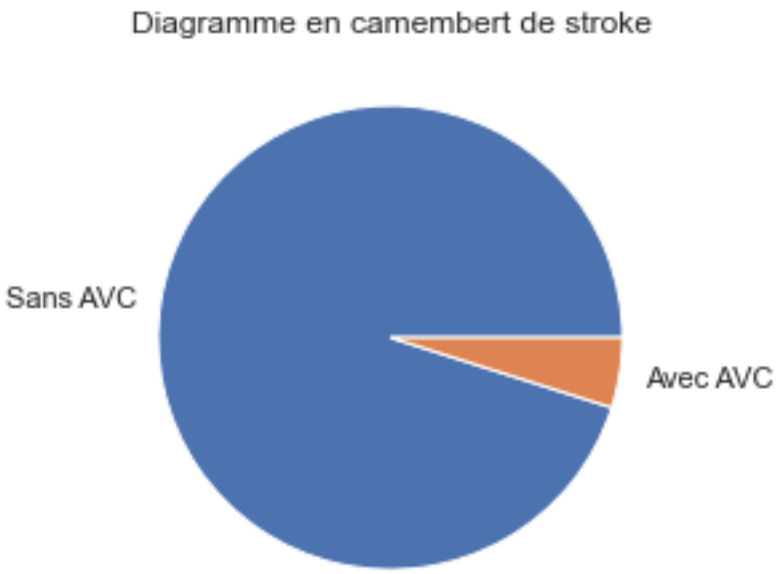
Nous devons voir la proportion de chaque classe au sein de la variable cible pour identifier si on a affaire à un désiquilibre de classes.

Proportion de chaque classe.

```
0    95.127202
1     4.872798
Name: stroke, dtype: float64
```

Diagramme en camembert de la proportion de chaque classe.

```
Text(0, 0.5, '')
```



Nous remarquons que 95.13% des patients sont atteints d'AVC contre seulement 4.87% de patients non atteints. Nous notons donc un fort désiquilibre de classes.

2.7 Compréhension des différentes variables

Analysons les variables à l'aide de graphiques.

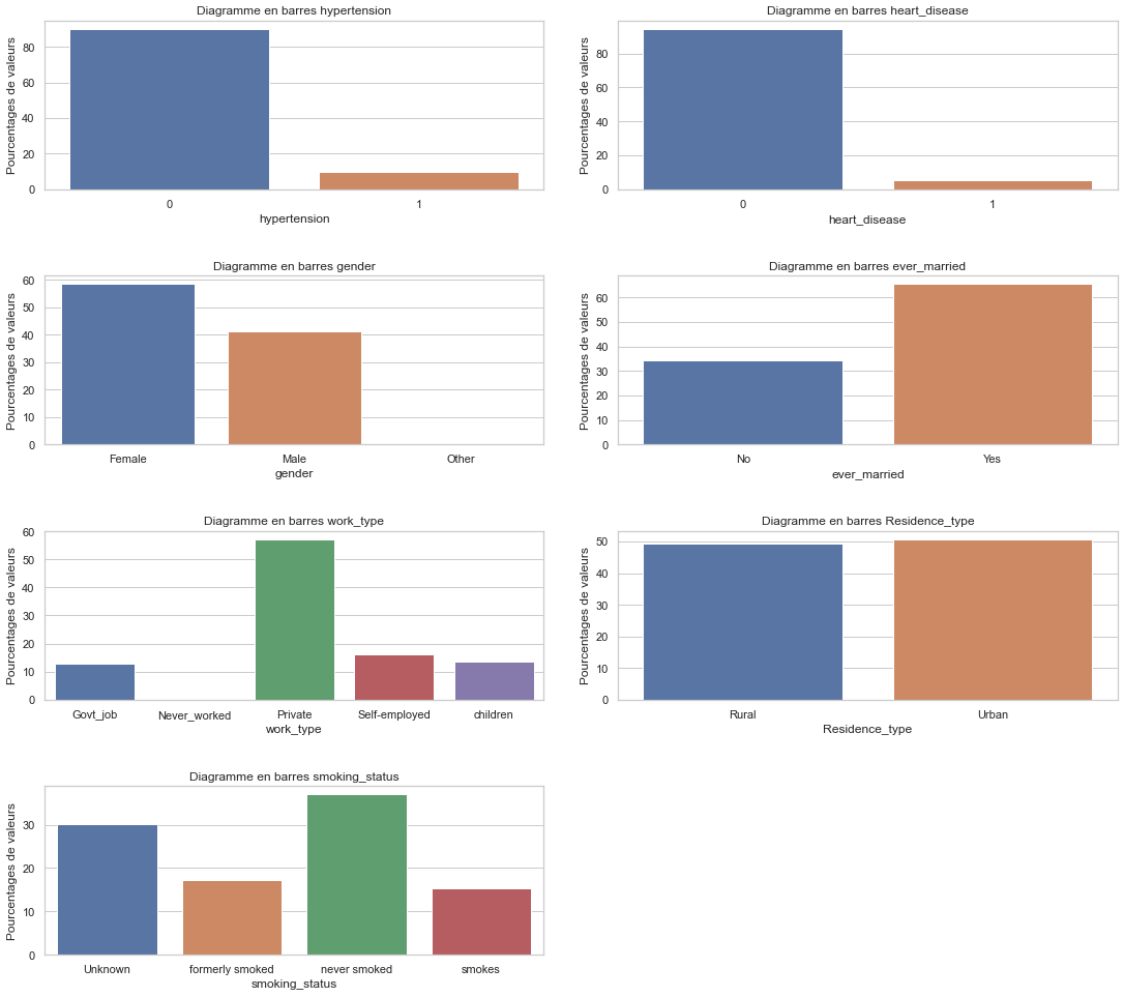
2.7.1 Variables catégorielles

Proportions des modalités pour chaque variable catégorielles (en pourcentages).

```
Colonne hypertension :
0    90.254403
1     9.745597
Name: hypertension, dtype: float64
-------------------

Colonne heart_disease :
0    94.598826
1     5.401174
Name: heart_disease, dtype: float64
-------------------

Colonne gender :
Female    58.590998
Male      41.389432
Other      0.019569
Name: gender, dtype: float64
-------------------

Colonne ever_married :
Yes    65.616438
No     34.383562
Name: ever_married, dtype: float64
-------------------

Colonne work_type :
Private          57.240705
Self-employed    16.027397
children         13.444227
Govt_job         12.857143
Never_worked      0.430528
Name: work_type, dtype: float64
-------------------

Colonne Residence_type :
Urban    50.802348
Rural    49.197652
Name: Residence_type, dtype: float64
-------------------

Colonne smoking_status :
```

```
never smoked       37.025440
Unknown            30.215264
formerly smoked    17.318982
smokes             15.440313
Name: smoking_status, dtype: float64
--------------------
```

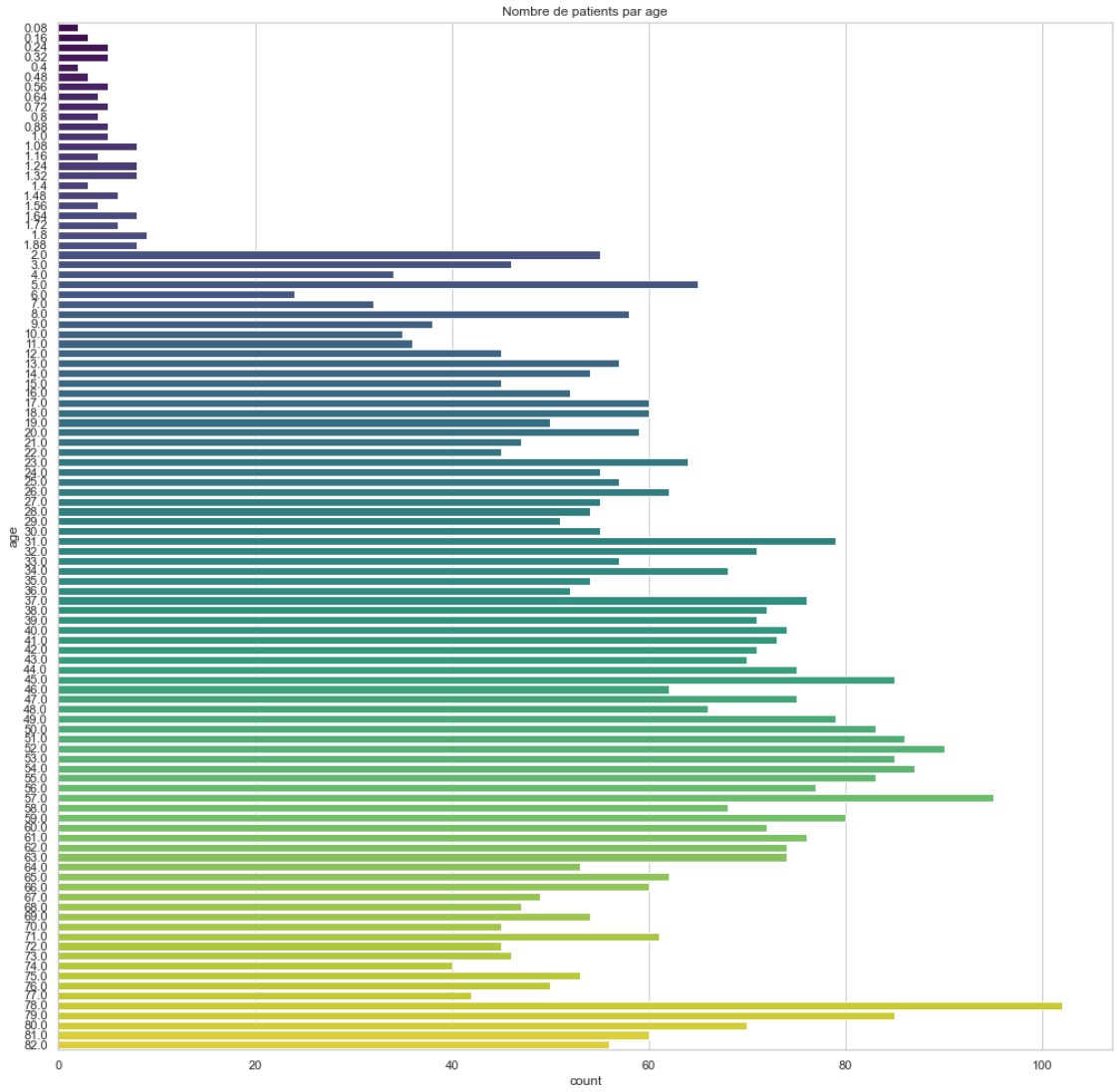
Diagramme en barres des proportions pour chaque variable.



2.7.2 Variables non catégorielles

Diagramme du nombre de patients en fonction de l'âge.

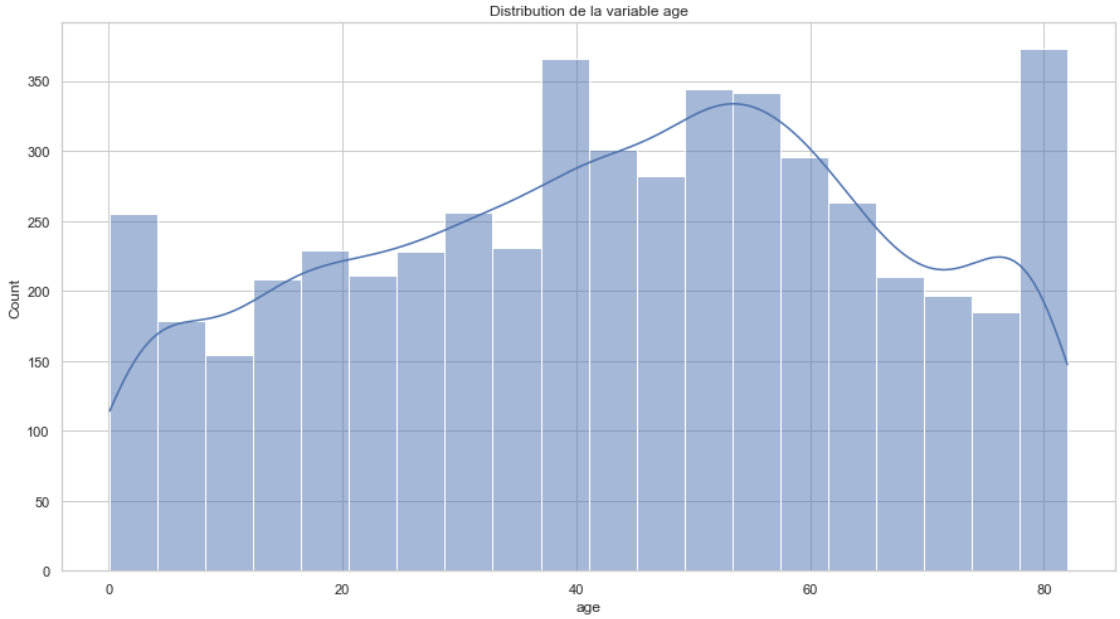
```
Text(0.5, 1.0, 'Nombre de patients par age')
```



Nous constatons qu'il y a plus de patients adultes que de patients jeunes. Le plus grand nombre de tests a été effectué sur les personnes agées de 78 ans.

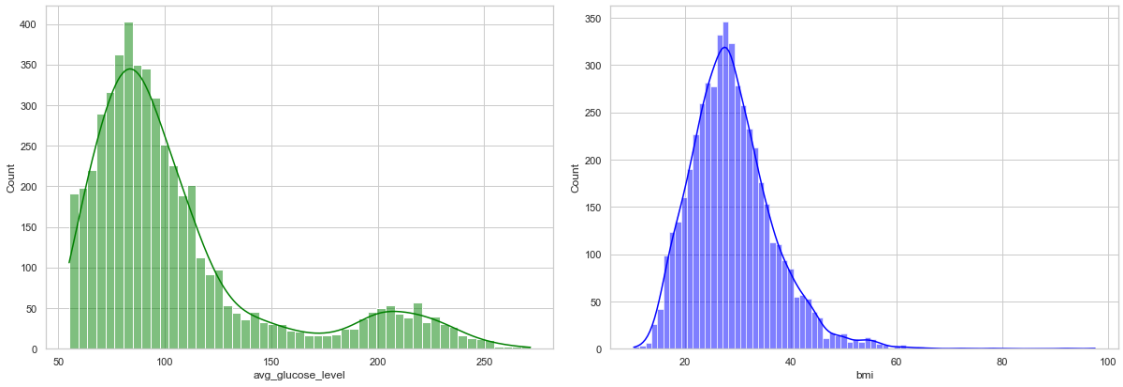
Distribution des âges.

```
Text(0.5, 1.0, 'Distribution de la variable age')
```



La variable age ne suit pas une distribution normale. Nous remarquons que la distribution est plus élevée au niveau des patients âgés entre 37 et 63 ans. Elle est moins élevée au niveau des jeunes patients.

histogrammes du *niveau moyen de glucoses dans le sang* (avg_glucose_level) et de l'*indice de masse corporelle*(bmi).



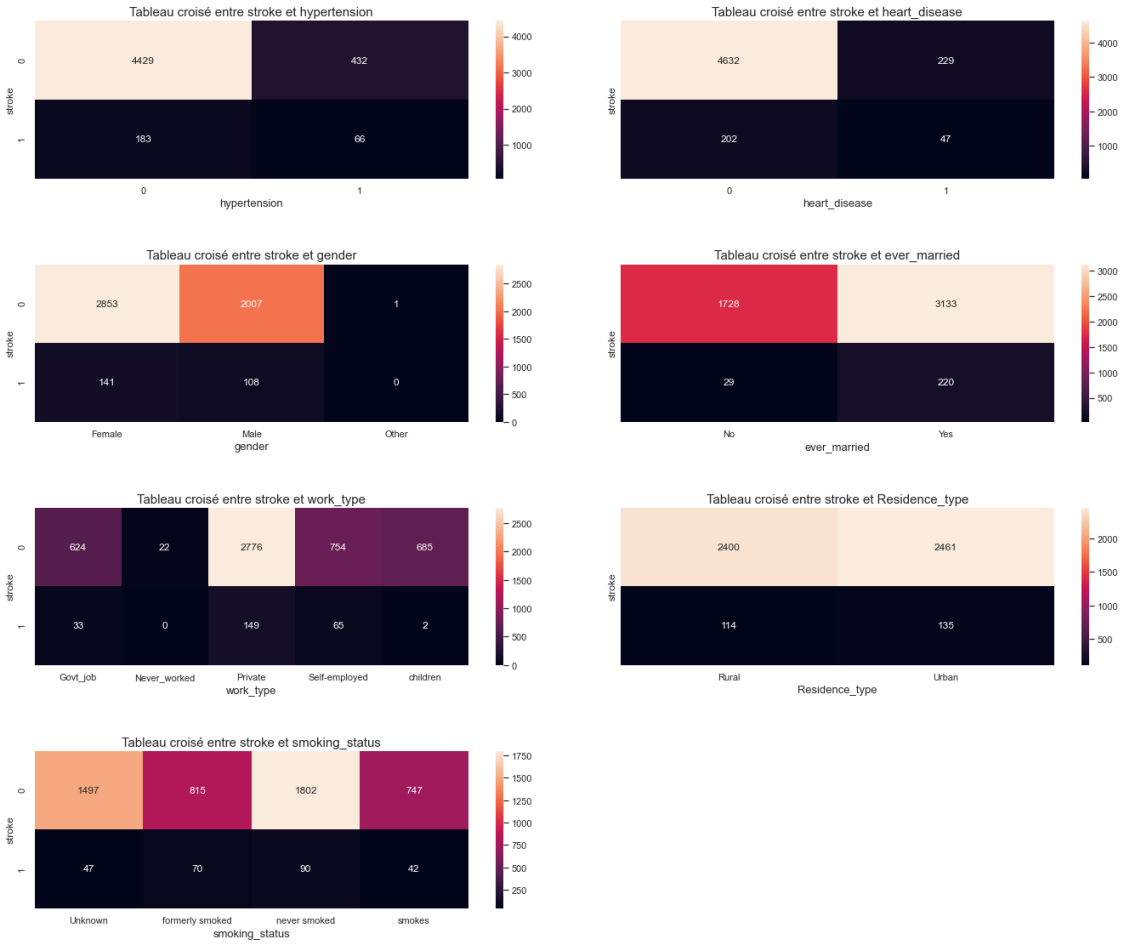
Nous remarquons, pour la variable avg_glucose_level, une composition de deux distributions (la deuxième à l'aire de se détacher de la première). La première distribution ressemble à une distribution normale avec une plus grande variance que la deuxième et sa moyenne tourne autour de 75-87 mg/dL de sang. Pour la deuxième distribution, dont la variance est plus faible, nous notons une moyenne tournant autour de 210-225 mg/dL.

Nous remarquons, pour la variable bmi, une distribution qui semble être normale avec une moyenne tournant autour de 30 kg/m2.

2.8 Relations entre la cible et les variables explicatives

2.8.1 Variables catégorielles / cible

Analysons ces relations en utilisant les tableaux de contingence entre la cible et chaque variable catégorielle.

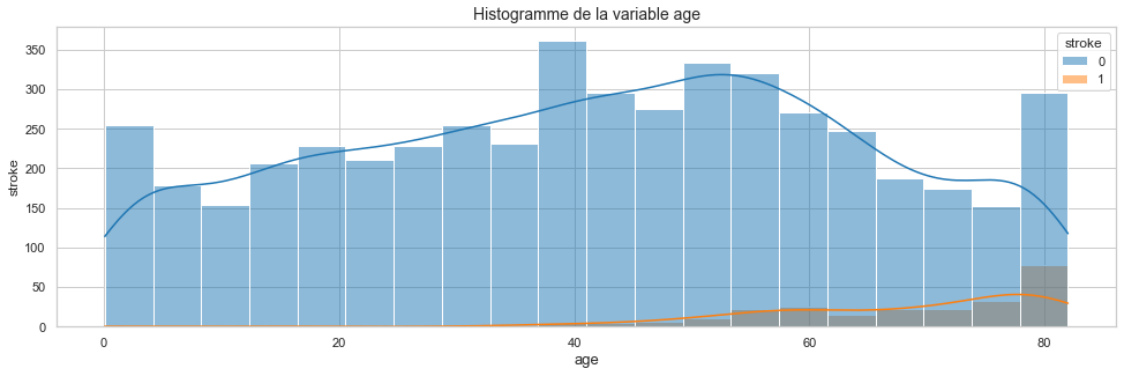


Nous remarquons que pour la variable work_type, on a une proportion de patients malades moins importante au niveau de la catégorie children. Nous remarquons également que pour la variable smoking_status, la proportion de patients malades est moins importante au niveau de la catégorie unknown. Pour le reste des variables nous ne notons pas de différences considérables entre les différentes proportions.

2.8.2 Variables non catégorielles / variable cible

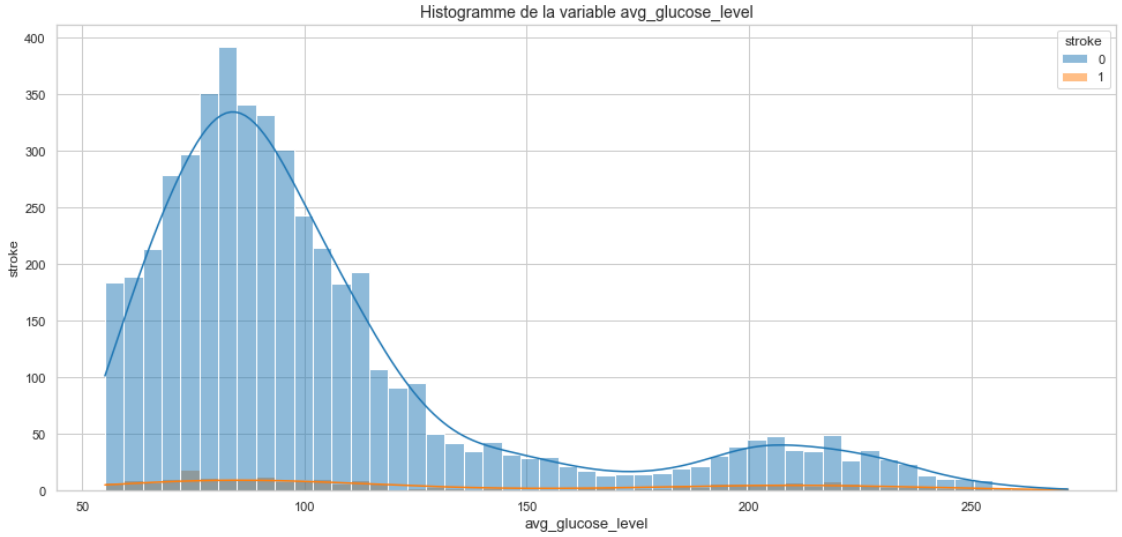
Dans cette partie, nous devons affiner les analyses effectuées sur les variables non catégorielles en traçant des distributions suivant les différentes classes possibles de la variable cible. Nous vérifierons ensuite si les distributions obtenues sont semblables.

Pour la variable age



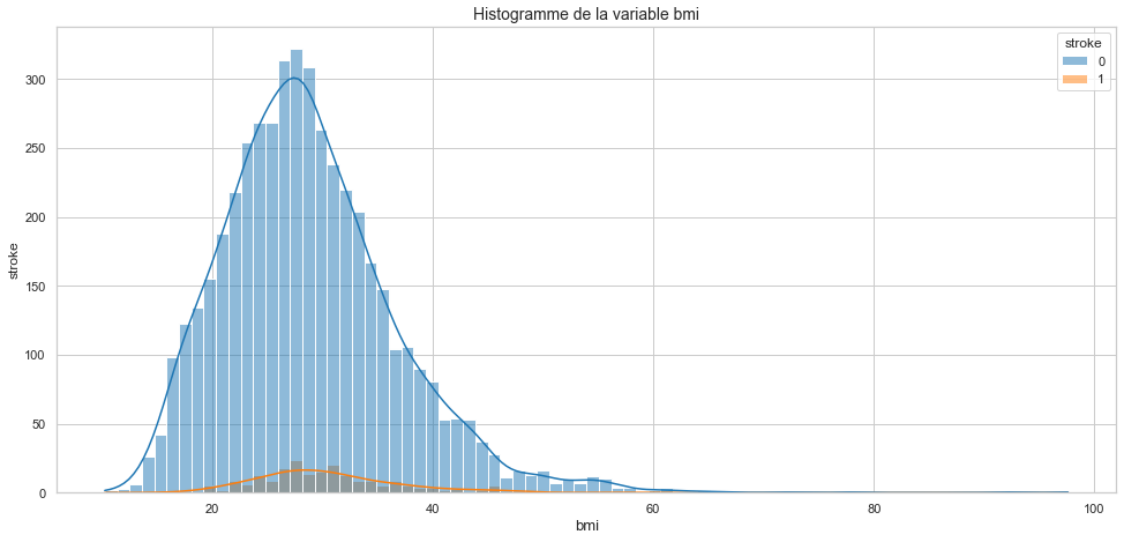
Nous voyons que les deux distributions sont différentes. Nous constatons que les personnes plus agées ont plus de risque d'être atteintes d'AVC.

Pour la variable avg_glucose_level



Nous remarquons une différence entre les deux distributions qui ne semble pas très visible.

Pour la variable bmi. Nous pouvons remplacer les valeurs manquantes de la variable par la valeur la plus fréquente (mode).

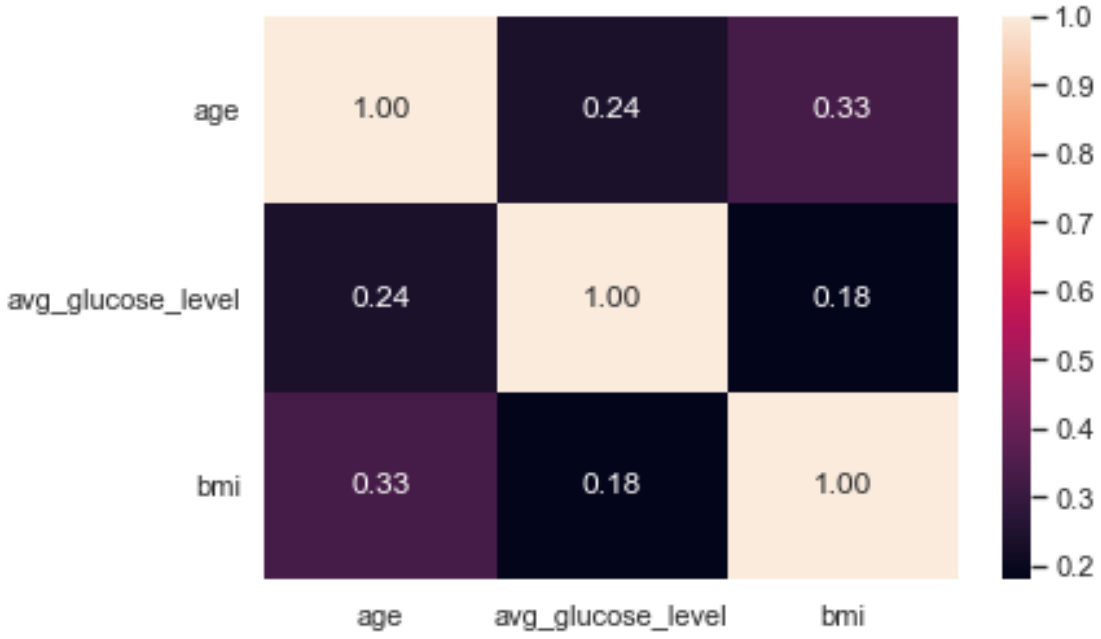


Nous obtenons à peu près les mêmes distributions pour les deux classes de la cible.

2.9 Relations entre variables non catégorielles

Graphique de corrélations entre les variables quantitatives.

```
<AxesSubplot:>
```

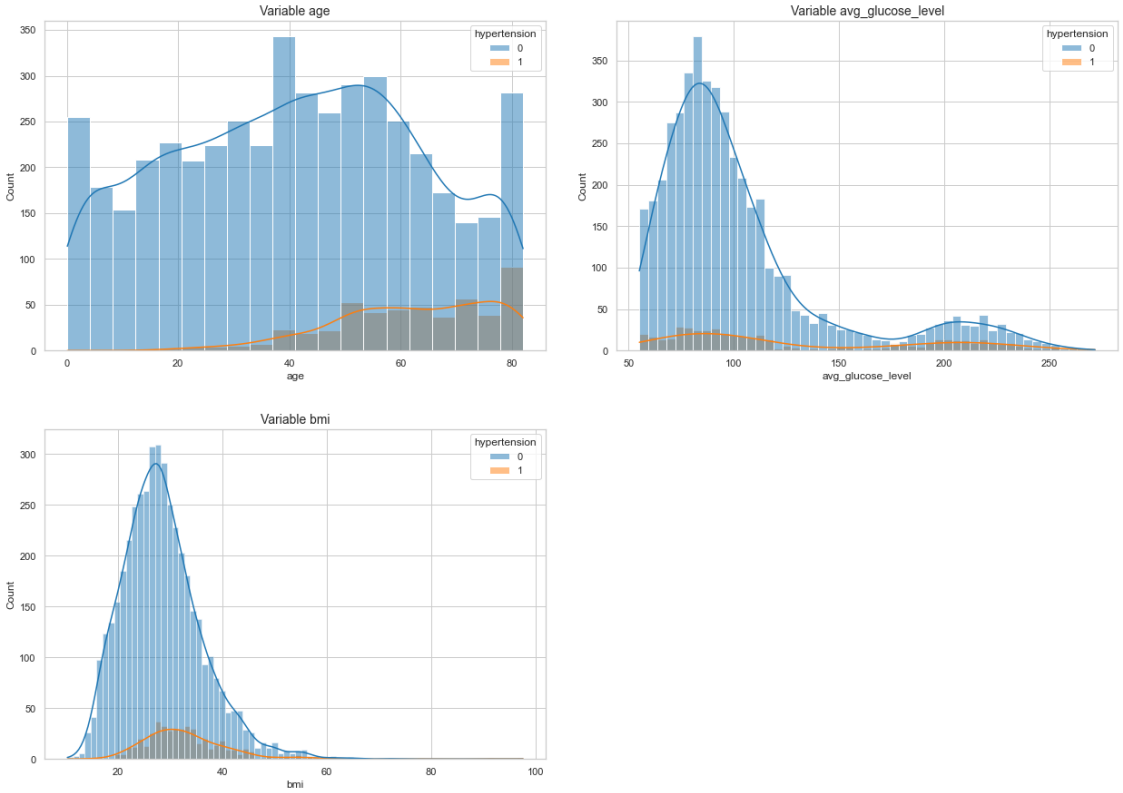


Nous remarquons qu'aucune des coefficients de corrélation obtenus ne dépasse 50%. Les variables quantitatives ne partagent aucune forte corrélation entre elles.

2.10 Relations entre variables catégorielles et non catégorielles

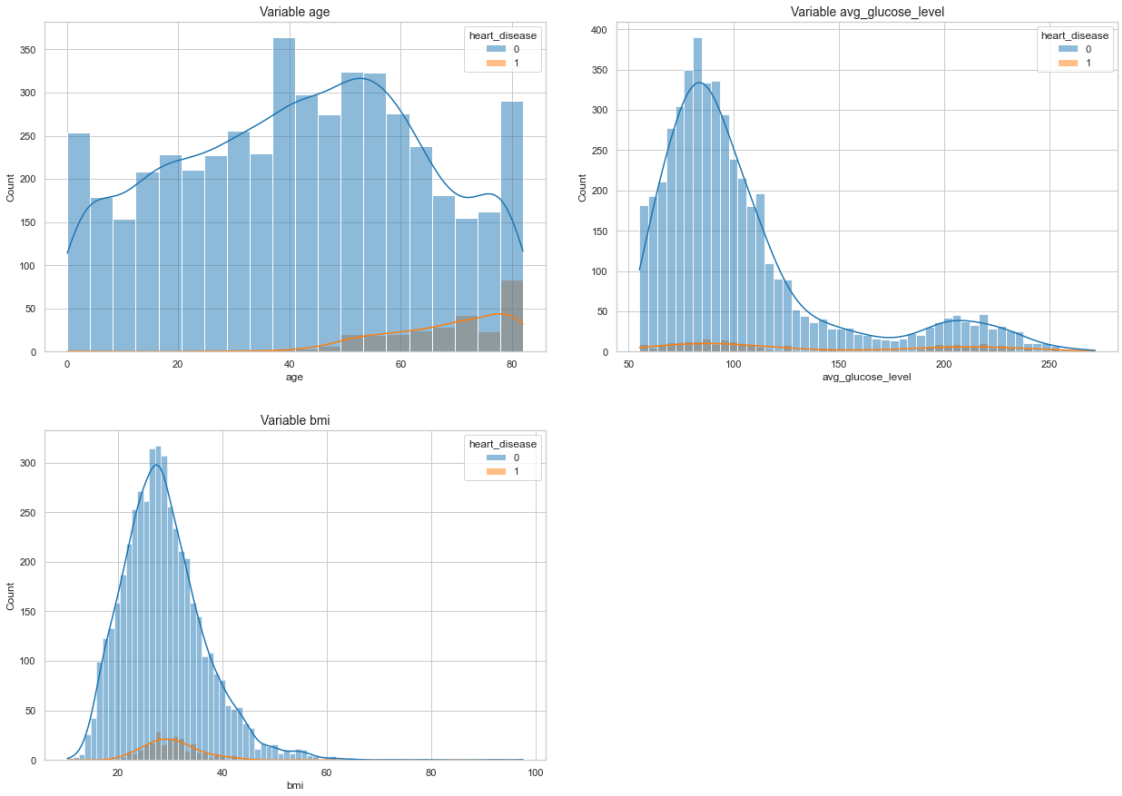
Nous allons vérifier pour chaque variable quantitative si elle est influencée par une ou plusieurs variable(s) catégorielle(s).

Pour la variable hypertension



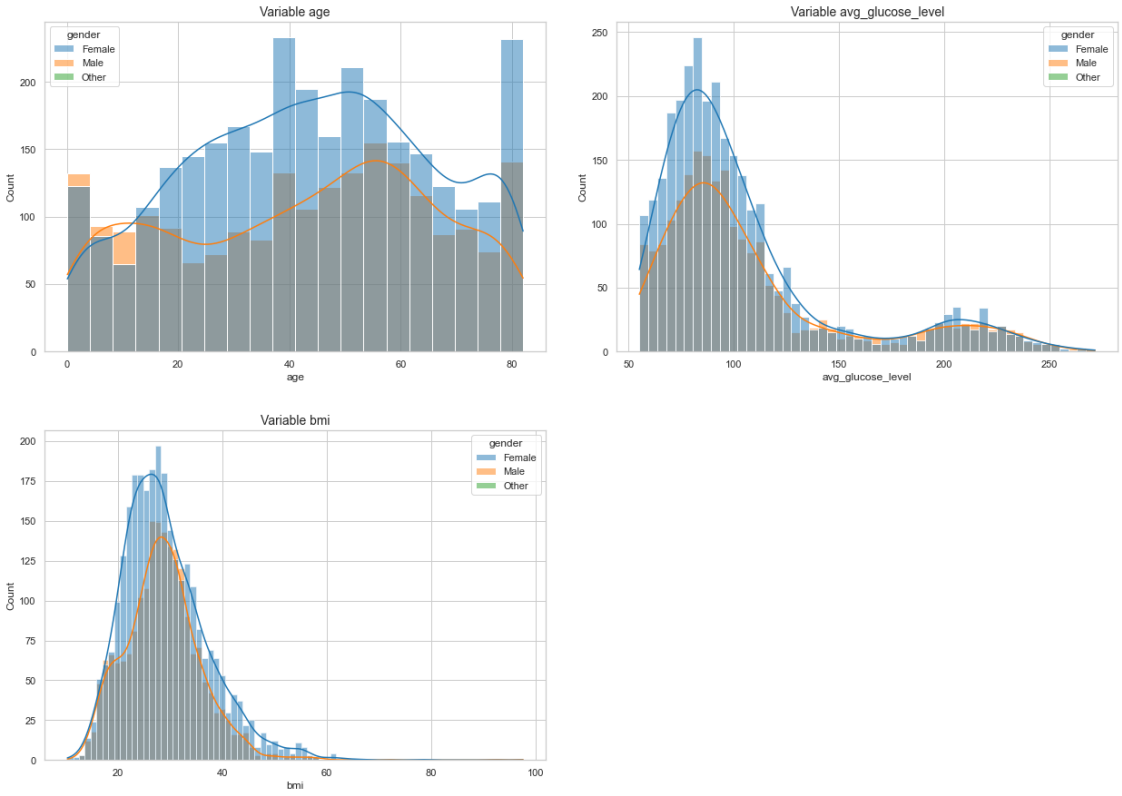
Nous remarquons une **légère dépendance** entre les variables **hypertension** et **age**

Pour la variable heart_disease



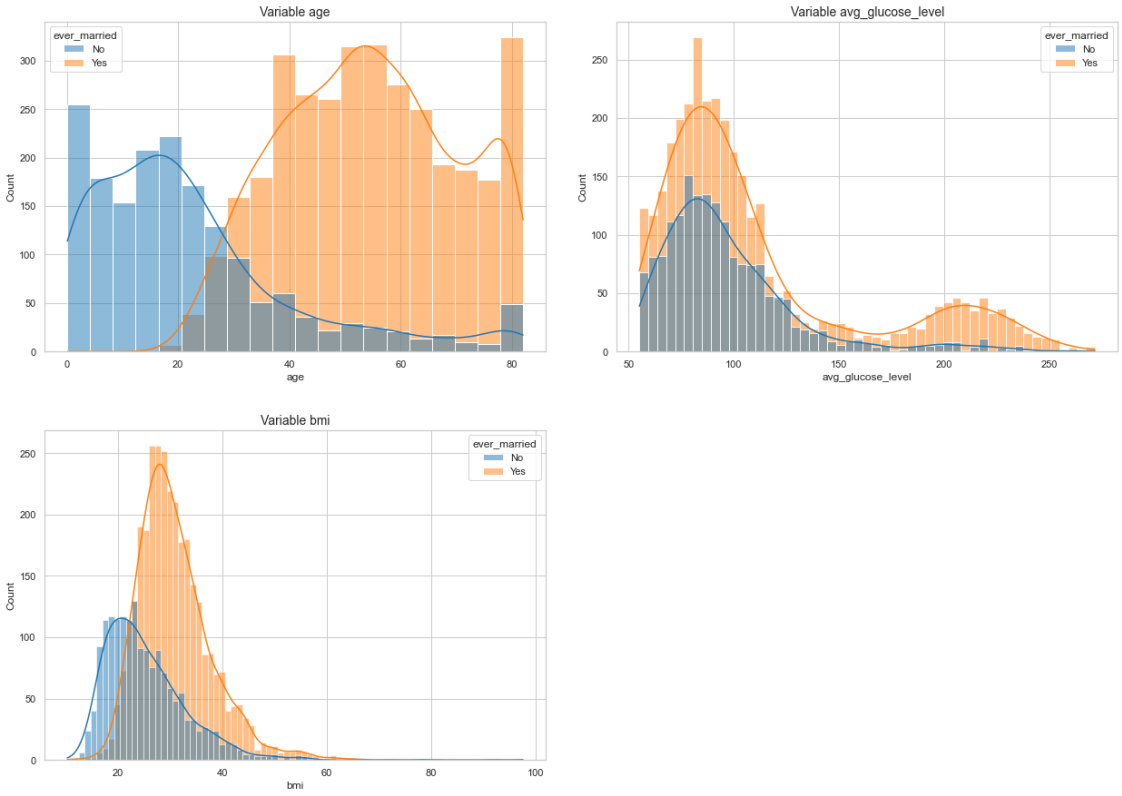
Nous constatons également une légère dépendance entre les variables heart_disease et age.

Pour la variable gender



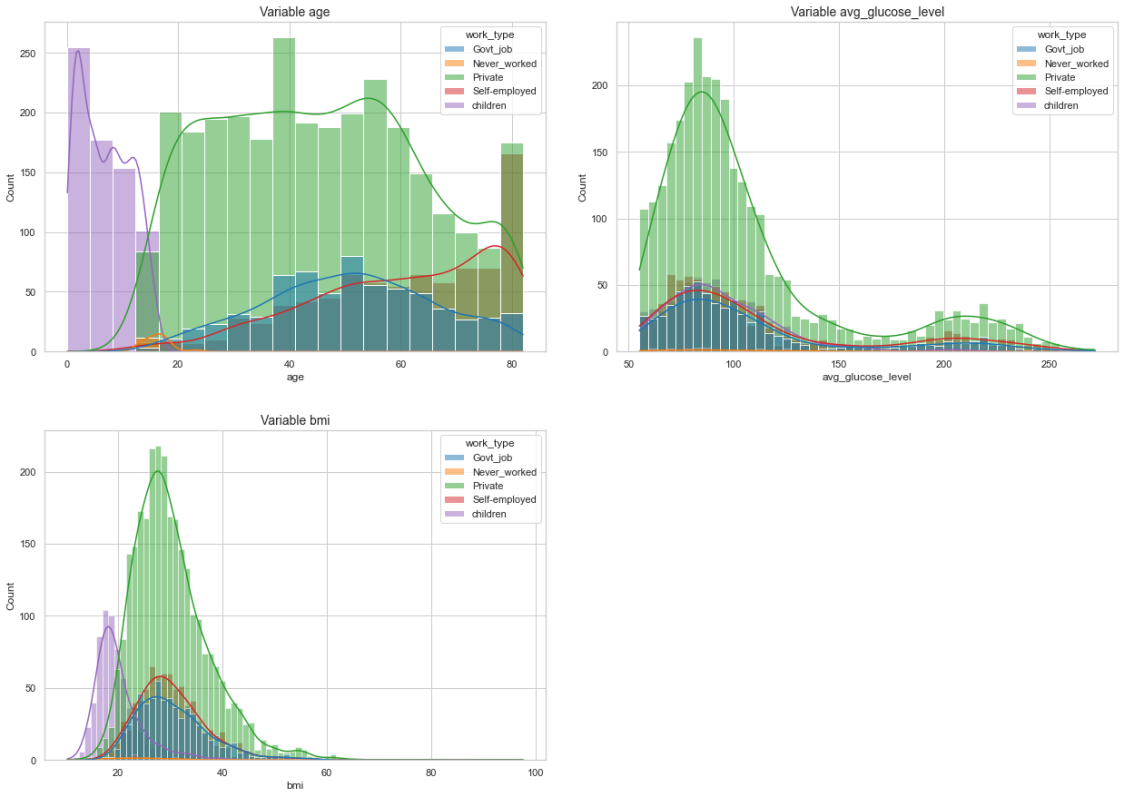
Il n'existe **aucune forte corrélation** entre la variable **age** et **gender**

Pour la variable ever_married



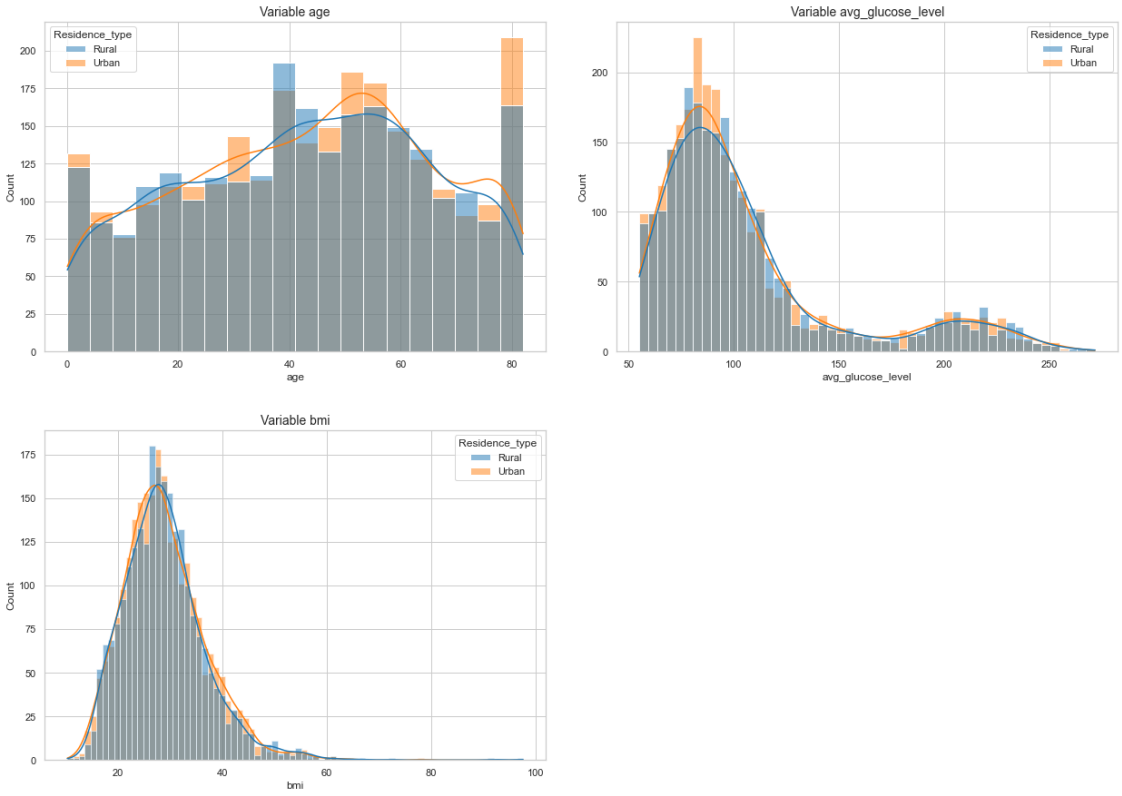
La variable **ever_married** est **fortement corrélée** avec les variables **age** et **bmi**

Pour la variable work_type



La variable **work_type** également réalise une **forte corrélation** entre les variables **age** et **bmi**. *Nous remarquons que la catégorie children de work_type désigne les patients enfants (donc la variable work_type est cohérente pour le reste de l'analyse).*

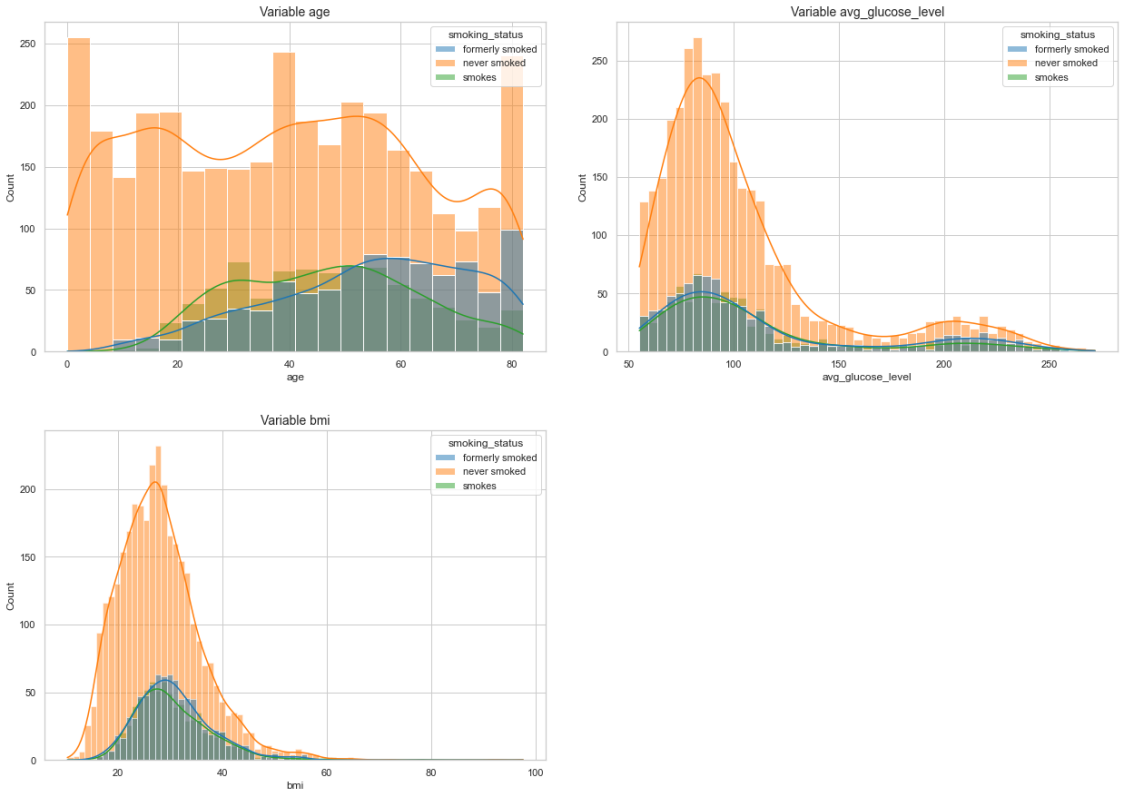
Pour la variable residence_type



La variable **Residence_type n'est pas corrélée** avec la variable **age**.

Pour la variable smoking_status. Remplacons la catégorie *unknown* par *never smoked* pour avoir une meilleure représentation de la variable.

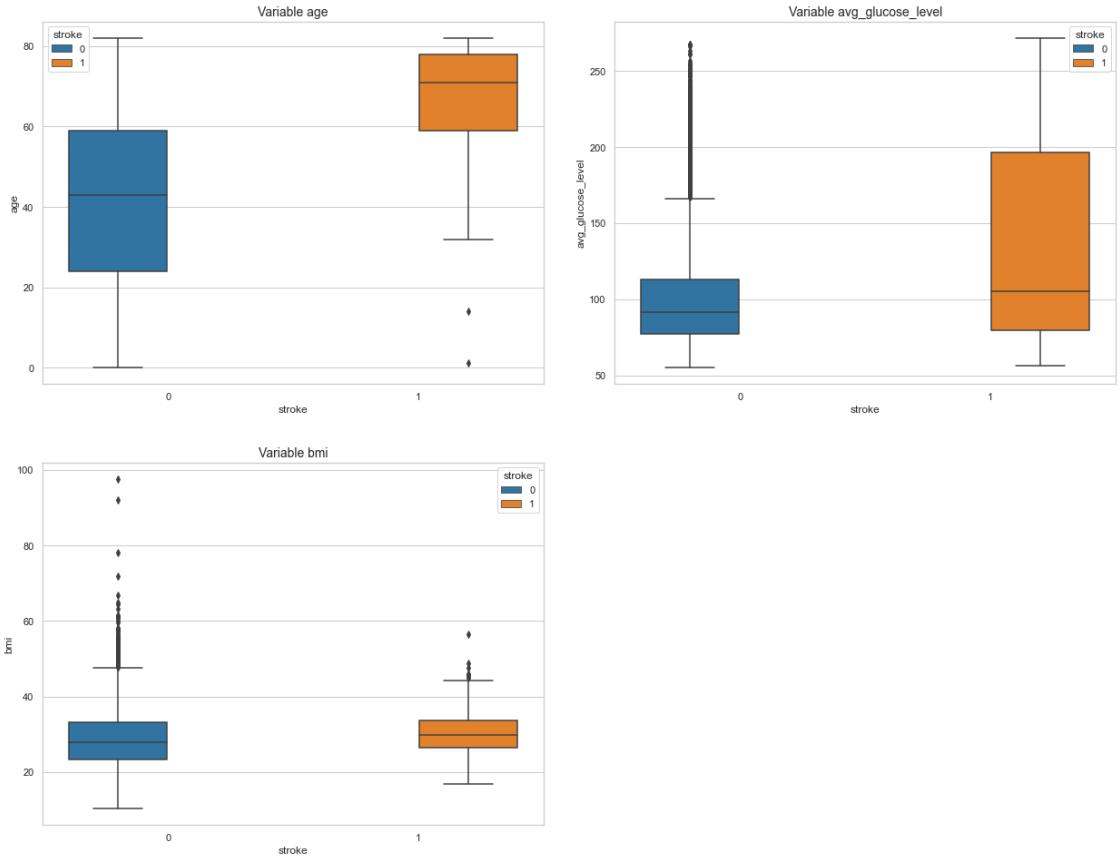
```
never smoked       67.240705
formerly smoked    17.318982
smokes             15.440313
Name: smoking_status, dtype: float64
```



La variable **smoking_status** est **fortement corrélée** avec la variable **age**.

2.11 Identification des valeurs aberrantes

Diagrammes en moustache des variables catégorielles par classe de patients (atteints d'AVC ou pas).



Nous constatons que tous les trois variables comportent des valeurs 'aberrantes'. Surtout du coté de la classe *test negatif* de la variable stroke (pour les variables bmi et avg_glucose_level).

2.12 Test d'hypothéses

Vérifions si certaines variables quantitatives influencent le fait qu'un patient soit atteint d'AVC ou pas à l'aide d'un test de student (Hypothèse H0)

Nous obtenons les résultats suivants.

```
age--------------------------- : H0 rejetée
avg_glucose_level------------- : H0 rejetée
bmi--------------------------- : H0 non rejetée
```

Selon le test, seule la variable bmi n'entretient pas de relation avec la cible.

3 Phase de Prétraitement des données

3.1 Étapes du prétraitement

Nous allons effectuer plusieurs traitements sur les données et vérifier le score obtenu après entraînement d'un modèle d'arbre de décision (il ne s'agit pas du modèle final). Nous considérons

que l'arbre de décision est le modèle de classification le plus facile à interpréter c'est pourquoi il est nécessaire de l'utiliser durant la phase de prétraitement.

- **Suppression de variables** : Nous diminuons considérablement le risque de surentraînement de nos modèles en supprimant les variables inutiles. Les variables fortement corrélées entre elles fournissent les mêmes informations au modèle. Ce surplus d'informations peut être corrigé avec la suppression de certaines variables qui sont fortement corrélées avec d'autres.
- **Suppression des données aberrantes** : Toutes les valeurs de la variable avg_glucose_level supérieures à environ 169 mg/dL, soit les valeurs désignant les patients qui ont un taux anormal de glucose dans le sang, sont supprimées. Nous obtenons ainsi une distribution à peu près normale pour la variable avg_glucose_level (La deuxième distribution est complètement supprimée). La distribution de la variable age reste à peu près la même. Après la suppression des valeurs aberrantes, la taille de l'échantillon est de **4483** observations. Cependant les variables considérées comme anormales peuvent être importantes lorsque nous aurons un nombre de malades plus grand dans la base de données (grâce au sur échantillonnage). Cela nous permettra notamment d'avoir un meilleur score pour la classe 1 qui était considérée comme un bruit à cause du fort déséquilibre de classes. Nous avions retenu en effet, au cours de la phase d'exploration, qu'un taux de glucose élevé dans le sang désigne les patients atteints de diabète.
- **Séparation des données** en données d'entraînement et de test : Les données d'entraînement vont contenir **3586** observations et les données de test, **897** observations.
- **Encodage des données qualitatives** : Il est important d'encoder les données de type *chaîne de caractères* avant de procéder à l'entraînement car les modèles qui seront utilisés sont incapables d'effectuer des calculs sur des données qualitatives sans une préalable transformation.
- **Entraînement et évaluation du modèle de test** : En supprimant uniquement les variables bmi, work_type, ever_married, smoking_status et id on obtient une sensibilité de **96%** pour la classe *test negatif* et seulement **20%** pour la classe *test positif*.
- **Sélection de variables** : Les variables gender, hypertension, heart_disease et Residence_type sont celles qui doivent être supprimées. Ces variables sont les moins explicatives du lot de variables qui ont servi au précédent entraînement. Les variables age et avg_glucose_level sont à l'inverse les meilleures variables.
- **Entraînement et évaluation du modèle de test** avec les variables choisies : Nous obtenons une sensibilité de **96%** pour la classe *test négatif* et **29%** (soit une amélioration de 9%) pour la classe *test positif*.
- **Sur échantillonnage des données** : On choisit un pourcentage de 50% pour la stratégie de sur échantillonnage de la classe minoritaire. On améliore considérablement la sensibilité de la classe minoritaire ; mais cette stratégie diminuera en contrepartie la spécificité. En décidant de ne pas supprimer les valeurs 'anormales' de la variable avg_glucose_level, nous allons encore plus améliorer la sensibilité de la modalité *test positif* et, en contrepartie, diminuer la spécificité.
- **Entraînement et évaluation du modèle de test** après sur échantillonnage : Les distributions obtenus sont plus distinctives que celles d'avant. Par contre la distribution normale qu'on avait au niveau de la variable avg_glucose_level a été perdu. Cela risque de détéri-

orer la spécificité mais en contrepartie on aura une meilleure sensibilité pour la modalité *test positif*.

- **Conclusion** : Nous obtenons un bon score pour la classe *test négatif* mais pas pour la classe *test positif* qui reste très faible. Nous supposons que les données utilisées n'expliquent pas bien le phénomène *attraper un AVC*. Nous devons vérifier par la suite si on obtiendra de meilleures sensibilité et aire sous la courbe avec des modèles plus performants que l'arbre de décision. Grâce au sur-échantillonnage nous allons pouvoir éliminer les erreurs d'apprentissage qui seront commises par les modèles qui seront utilisés par la suite. On décide d'éliminer les valeurs aberrantes pour obtenir une meilleure aire sous la courbe que si l'on avait uniquement privilégier la sensibilité.
- **Recommandations** : Il est crucial de recueillir plus de données et d'augmenter le nombre de variables pertinentes nécessaires à l'analyse et à l'entraînement d'un modèle de détection d'AVC. La complétude dans la collecte n'a pas été totalement respectée. Une bonne stratégie de lutte contre l'AVC nécessite un supplément de variables très utiles à sa *prévention primaire*, telles que les suivantes :
  - **Le taux de cholestérol dans le sang** (en g/L) : Le cholestérol est un corps gras indispensable au fonctionnement de l'organisme. Il entre notamment dans la composition des membranes des cellules et sert, entre autres, de **' matière première '** à la synthèse de nombreuses hormones (stéroïdes). Cela étant, le cholestérol en excès peut être nuisible, car il a tendance à s'accumuler dans les vaisseaux sanguins et à former des plaques dites d'athérosclérose qui peuvent, à terme, augmenter le risque cardiovasculaire. L'analyse de cette variable nous permettrait par exemple de connaître le profil lipidique du patient (s'il est en état d'excès de cholestérol ou pas).
  - **La fibrillation auriculaire** : La fibrillation auriculaire (ou atriale) est un trouble du rythme cardiaque qui accélère le cœur et le fait battre de manière irrégulière. Son apparition est favorisée par le vieillissement et la présence d'une pathologie cardiaque (HTA, maladie des valves cardiaque…), d'une obésité, d'un syndrome d'apnée du sommeil obstructive etc.
  - **Consommation d'alcool du patient** : Cette variable pourrait avoir plusieurs valeurs possibles -
    - Le patient ne consomme pas d'alcool ;
    - Le patient consomme une certaine quantité par occasion (ex : cinq verres par occasion au moins une fois par semaine) ;
    - Le patient consomme quotidiennement de l'alcool.

  Cela nous permettra de savoir si le patient a une habitude de consommation qui dépasse les limites de ce qui est considéré comme une consommation "modérée" ou "socialement acceptable". Cette variable peut aussi être exprimée en termes de quantités (litres) consommées par jour/semaine/mois/…
  - **La sédentarité** : elle correspond à une activité physique faible ou nulle avec une dépense énergétique proche de zéro ; la mesure du temps passé devant un écran que ce soit un ordinateur ou une télévision est un très bon indicateur de sédentarité.

  Des données supplémentaires peuvent également être recueillies en plus de celles fournies ci-dessus :

- Vérifier si le patient a une fibrillation musculaire (une forme d'arythmie cardiaque) ;
- Vérifier si la personne a déjà eu un accident ischémique transitoire (mini-AVC) ou un AVC ;
- Vérifier si la personne a un nombre élevé de globules rouges dans le sang (polyglobulie) ;
- Vérifier si la personne a un proche parent qui a été atteint d'un AVC ;
- indiquer le statut matrimonial du patient ainsi que le nombre d'enfants qu'il a.

Plein d'autres variables, non précisées ici, peuvent être ajoutées au jeu de données. Il est fort possible que certaines d'entre elles soient corrélées ou qu'elles puissent apporter beaucoup d'informations au modèle choisi. Quoi qu'il en soit, il sera nécessaire de recueillir correctement ces données (dans les normes) et d'effectuer un certain nombre de nettoyages approfondis sur ces dernières.

Il est possible de recueillir ces données à partir d'un site de consultation de dossiers médicales personnels de patients sur lesquels on a effectué des examens confirmant ou non la présence d'AVC. La consultation de ces sites peut nécessiter l'accord d'hôpitaux ou de clinique(s) en charges de ses dossiers médicaux. Un autre moyen d'acquérir ces données est d'effectuer une demande de consultation de dossiers personnels de patients, dans le but d'une enquête, à un professionnel de santé ou bien à l'accueil d'un établissement de santé (hôpitaux, cliniques...). Il peut nécessiter tout de même, à/aux enquêteur(s), d'acquérir un budget assez considérable pour effectuer cette enquête, sauf si elle est financée par une association ou autres, car souvent ces données sont à caractère personnel et peuvent être assujeties à des droits fondamentaux.

3.2 suppression de variables

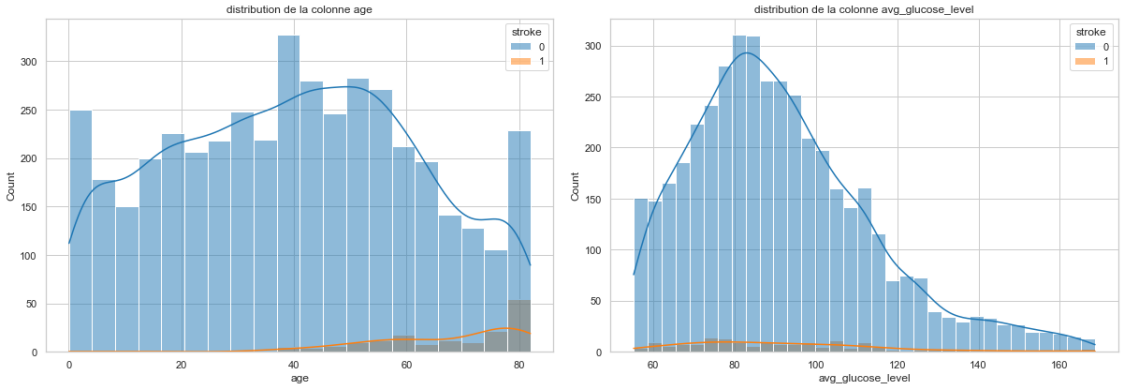
Nous devons supprimer les colonnes bmi, work_type, ever_married, smoking_status et id.

`Les variables restantes sont : gender, age, hypertension, heart_disease, Residence_type, avg_glucose_level, stroke`

3.3 Suppression des données aberrantes

Nous devons supprimer les valeurs aberrantes des variables age et avg_glucose_level.

Traçons les distributions des deux variables quantitatives pour vérifier les changements effectués.



```
Sans valeurs aberrantes, il nous reste  4483  observations dans le jeu de
données.
```

3.4 Encodage des données qualitatives

Le nouveau jeu de données ne contiendra plus que des données quantitatives. *Cette étape n'est pas forcément utile car aucune des variables restantes n'est de type catégoriel.*

3.5 Séparation des données

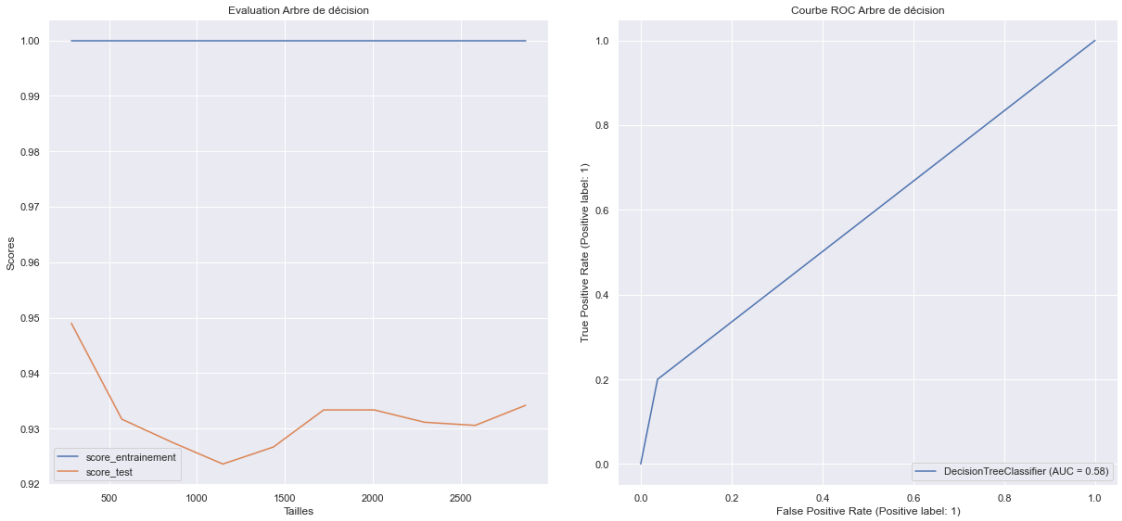
Vérifions les tailles des variables explicatives

```
Taille des données d'entraînement :  3586  observations
Taille des données de test :  897  observations
```

3.6 Premier entraînement avec un arbre de décision

Résultats du premier entraînement.

```
La precision du modèle : 0.9331103678929766
Matrice de confusion :
[[830  32]
 [ 28   7]]
Rapport de classification :
              precision    recall  f1-score   support

           0       0.97      0.96      0.97       862
           1       0.18      0.20      0.19        35

    accuracy                           0.93       897
   macro avg       0.57      0.58      0.58       897
weighted avg       0.94      0.93      0.93       897
```



Nous obtenons une sensibilité égale à 20% (<50%) et une aire sous la courbe de 58% (proche de 50%). Nous constatons également un sur entraînement du modèle. Eliminons les variables non informatives détectées pendant la phase d'exploration, notamment avec le tableau de contingence, à l'aide de technique de sélection de variables.

3.7 Sélection de variables

Vérifions quelles sont les variables qui apportent le plus d'information au modèle d'arbre de décision.

```
Colonnes choisies :
 ['age', 'avg_glucose_level']
```

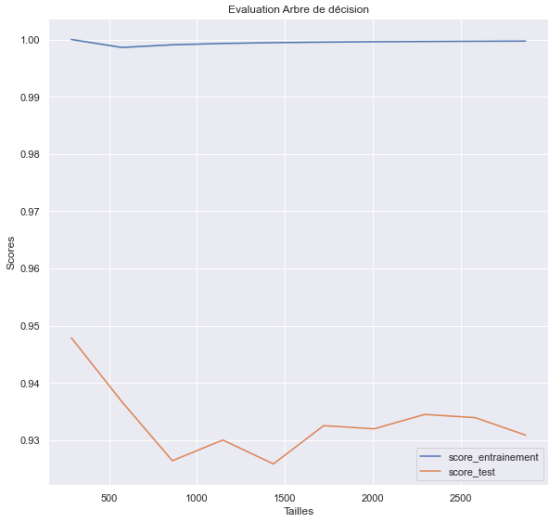
Les variables qui n'apportent pas d'information au modèle sont les variables gender, hypertension, heart_disease et Residence_type qui vont être supprimées des données d'entraînement et de test.

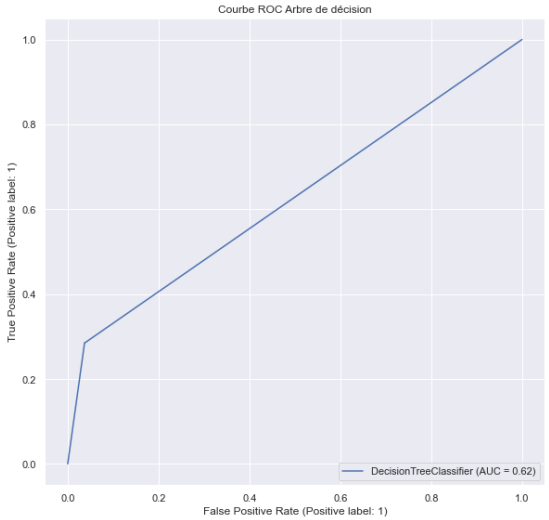
3.8 Deuxième entraînement du modèle avec les variables choisies

Résultats du deuxième entraînement.

```
La precision du modèle : 0.9364548494983278
Matrice de confusion :
[[830  32]
 [ 25  10]]
Rapport de classification :
              precision    recall  f1-score   support

           0       0.97      0.96      0.97       862
           1       0.24      0.29      0.26        35

    accuracy                           0.94       897
   macro avg       0.60      0.62      0.61       897
```

```
weighted avg       0.94      0.94      0.94       897
```

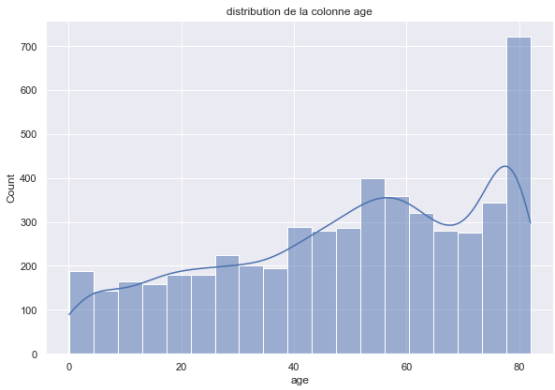


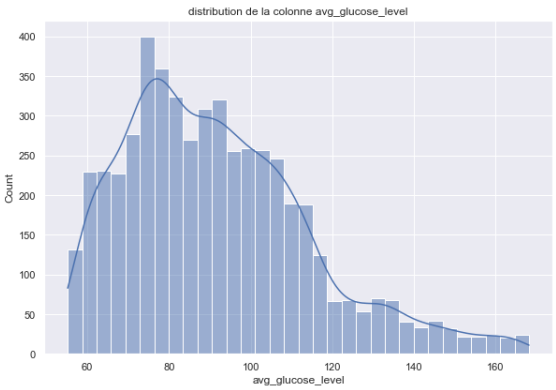


Nous notons une amélioration de la sensibilité de 9% ainsi que de l'aire sous la courbe qui passe à 62%. Nous notons toujours un sur entraînement du modèle mais moins grave que celui du premier entraînement. Par contre il est crucial d'augmenter la sensibilité. Pour cela nous pouvons utiliser des techniques de ré-échantillonnage. Le plus important est d'augmenter le nombre d'observations au niveau de la classe minoritaire. Le modèle pourra alors considérer la classe minoritaire comme une classe importante.

3.9 Sur échantillonnage des données

```
Comptage des classes avant sur échantillonnage :
Counter({0: 3456, 1: 130})
Comptage des classes après sur échantillonnage :
Counter({0: 3456, 1: 1728})
```

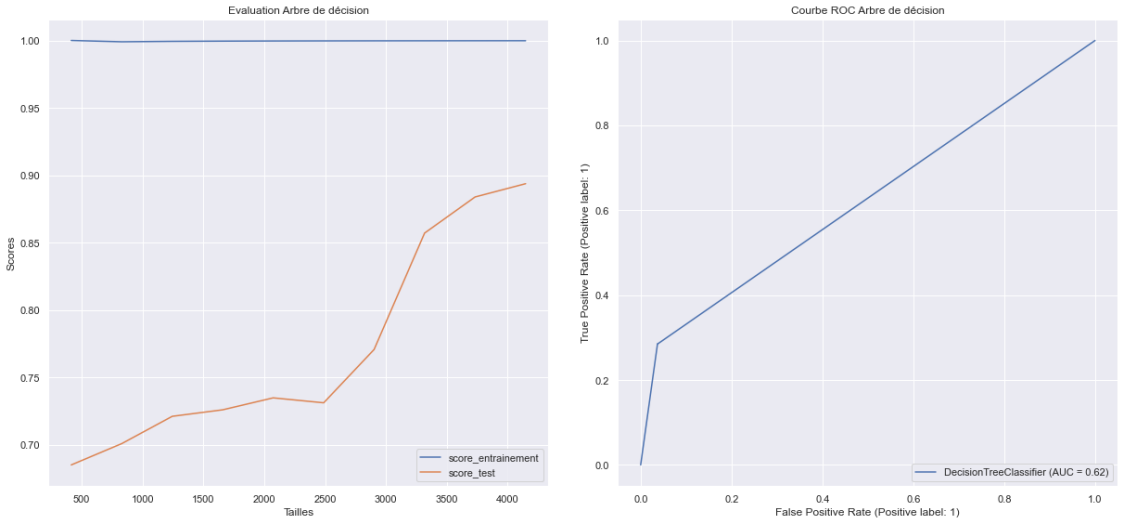




Les distributions obtenus sont plus distinctives que celles d'avant. Par contre la distribution normale qu'on avait au niveau de la variable avg_glucose_level a été perdue. Cela risque de détériorer la spécifité mais en contrepartie on aura une meilleure sensibilité pour la modalité *test positif*.

3.10 Troisième entraînement après sur échantillonnage

```
La precision du modèle : 0.9364548494983278
Matrice de confusion :
[[830  32]
 [ 25  10]]
Rapport de classification :
              precision    recall  f1-score   support

           0       0.97      0.96      0.97       862
           1       0.24      0.29      0.26        35

    accuracy                           0.94       897
   macro avg       0.60      0.62      0.61       897
weighted avg       0.94      0.94      0.94       897
```



Après le sur échantillonnage de la classe minoritaire, les scores obtenus avec l'arbre de décision ne change pas par rapport au dernier entraînement. Nous allons entraîner d'autres modèles plus performants que l'arbre de décision et vérifier si on obtient une amélioration des scores et une diminution du sur entraînement.

4 Phase de modélisation

4.1 Préambule

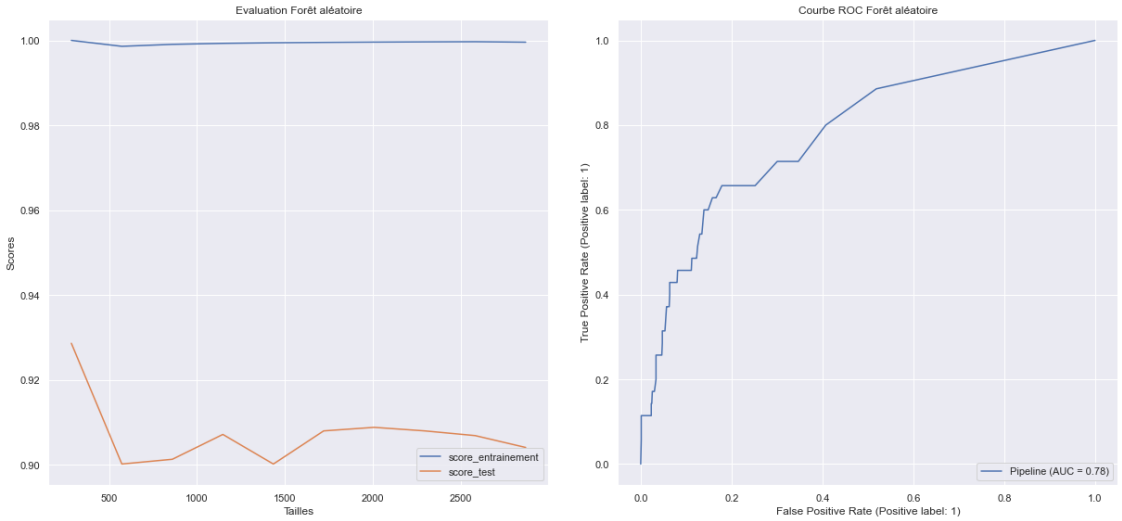
On devra entraîner un certain nombre de modèles et sélectionner celui qui donnera le meilleur score et qui ne sera ni trop complexe (score élevé sur les données d'entraînement et faible sur les données de test) et ni trop simple (score faible sur les données d'entraînement et de test). Le modèle final sera choisi parmi les suivantes : **RandomForestClassifier**, **AdaBoostClassifier**, **SVC**, **KNeighborsClassifier** et **LogisticRegression**.

L'entraînement d'un modèle nécessitera d'utiliser un pipeline (combinaison de traitements à effectuer) qui devra permettre **la discrétisation** des données quantitatives et **le sur échantillonnage** de la classe minoritaire avant **l'entraînement du modèle**. **L'évaluation du modèle** sera effectué immédiatement après son entraînement. *Pour la définition, discrétiser une variable quantitative c'est découper un vecteur de nombres réels en un vecteur de nombres entiers nommés "indices de classe" ou "codes de classe".*

4.2 Sélection de modèle

4.2.1 Random Forest

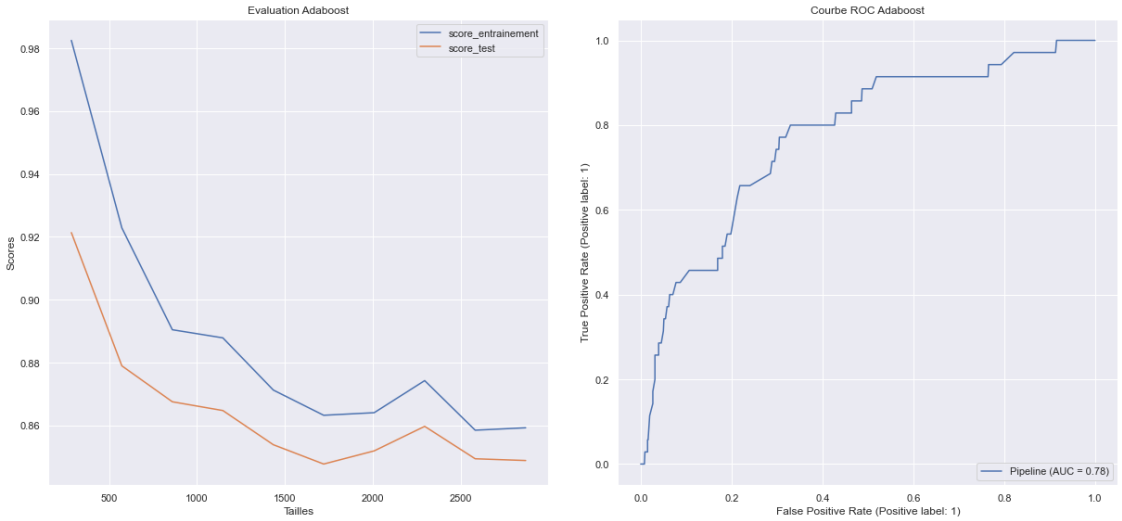
```
---------------
Pour le modèle Forêt aléatoire :
La precision du modèle : 0.9264214046822743
Matrice de confusion :
[[820  42]
 [ 24  11]]
Rapport de classification :
              precision    recall  f1-score   support

           0       0.97      0.95      0.96       862
           1       0.21      0.31      0.25        35

    accuracy                           0.93       897
   macro avg       0.59      0.63      0.61       897
weighted avg       0.94      0.93      0.93       897
```



Nous notons un sur entraînement du modèle. Nous obtenons une sensibilité de 31% et une aire sous la courbe de 78% soit, déjà, une grande amélioration par rapport au modèle de test.

4.2.2 Adaboost

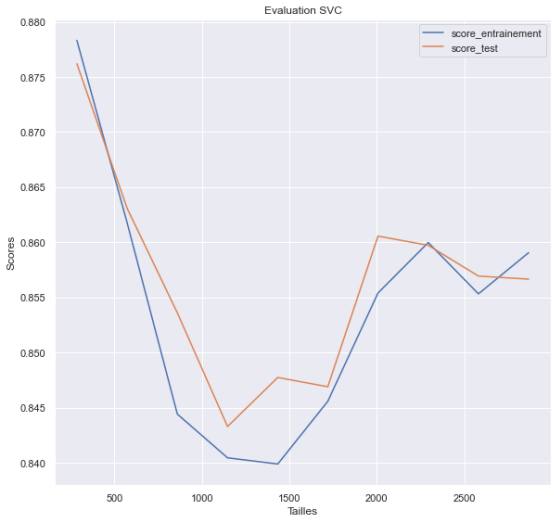
```
---------------
Pour le modèle Adaboost :
La precision du modèle : 0.830546265328874
Matrice de confusion :
[[729 133]
 [ 19  16]]
Rapport de classification :
              precision    recall  f1-score   support

           0       0.97      0.85      0.91       862
           1       0.11      0.46      0.17        35

    accuracy                           0.83       897
   macro avg       0.54      0.65      0.54       897
weighted avg       0.94      0.83      0.88       897
```

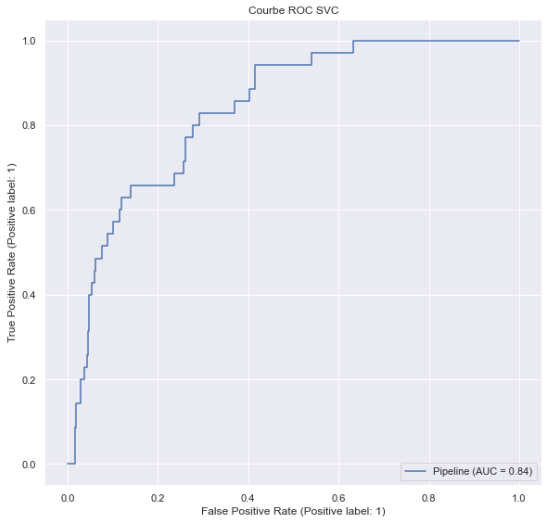


Ce modèle est plus performant que le précédent. On note une diminution du sur apprentissage et une amélioration de la sensibilité.

4.2.3 SVC

```
---------------
Pour le modèle SVC :
La precision du modèle : 0.8695652173913043
Matrice de confusion :
[[759 103]
 [ 14  21]]
Rapport de classification :
              precision    recall  f1-score   support

           0       0.98      0.88      0.93       862
           1       0.17      0.60      0.26        35

    accuracy                           0.87       897
   macro avg       0.58      0.74      0.60       897
weighted avg       0.95      0.87      0.90       897
```

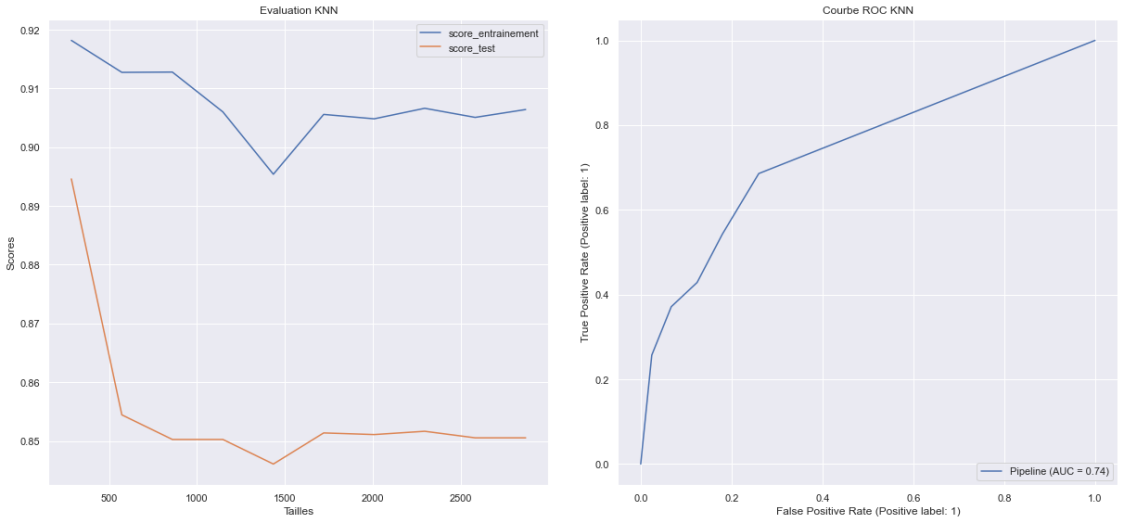




Ce modèle n'est quasiment pas sur entraîné et on obtient de meilleurs scores que pour les précédents modèles avec une AUC de 84% et une sensibilité de 60%.

4.2.4 KNN

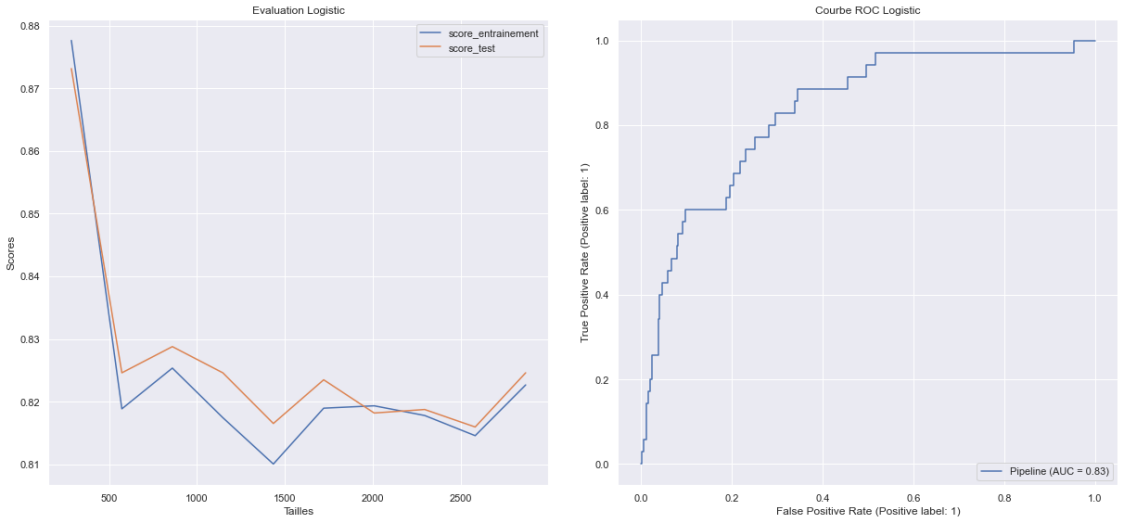
```
---------------
Pour le modèle KNN :
La precision du modèle : 0.8584169453734671
Matrice de confusion :
[[755 107]
 [ 20  15]]
Rapport de classification :
              precision    recall  f1-score   support

           0       0.97      0.88      0.92       862
           1       0.12      0.43      0.19        35

    accuracy                           0.86       897
   macro avg       0.55      0.65      0.56       897
weighted avg       0.94      0.86      0.89       897
```



La performance du modèle KNN est située entre celle du forêt aléatoire et de l'Adaboost. Le modèle est sur entraîné et on obtient une sensibilité de 43% et une aire sous la courbe de 74%.

4.2.5 LOGISTIC

```
---------------
Pour le modèle Logistic :
La precision du modèle : 0.8383500557413601
Matrice de confusion :
[[731 131]
 [ 14  21]]
Rapport de classification :
              precision    recall  f1-score   support

           0       0.98      0.85      0.91       862
           1       0.14      0.60      0.22        35

    accuracy                           0.84       897
   macro avg       0.56      0.72      0.57       897
weighted avg       0.95      0.84      0.88       897
```



Le meilleur modèle est la **régression logistique** avec une aire sous la courbe de **83%** et une sensibilité de **60%**. Bien que l'aire sous la courbe soit d'une unité inférieure à celle du modèle SVC on remarque que les scores obtenus pour les données d'entraînement et de test sont plus proches et montrent une meilleure harmonie.

4.3 Description du modèle choisi

4.3.1 C'est quoi la régression logistique ?

La régression logistique est une méthode qui permet de modéliser des variables binomiales (typiquement binaires), multinomiales (variables qualitatives à plus de deux modalités) ou ordinales (variables qualitatives dont les modalités sont ordonnées). Elle est très utilisée dans le domaine médical (guérison ou non d'un patient), en sociologie, en épidémiologie, en marketing quantitatif (achat ou non de produits ou services suite à une action) et en finance pour la modélisation de risques (scoring).

Le principe du modèle de la régression logistique est d'expliquer la survenance ou non d'un événement (la variable cible que nous noterons y) par le niveau de variables explicatives (notées X). Dans notre exemple, on cherche à prédire la sortie de la variable y (patient atteint d'AVC ou pas) en fonction des données relevées sur les patients.

4.3.2 Comment fonctionne la régression logistique ?

La régression logistique et la régression linéaire appartiennent à la même famille des modèles GLM **Generalized Linear Models**) : dans les deux cas, on relie un événement à une combinaison linéaire de variables explicatives.

Dans le cas de la régression linéaire ordinaire, la variable dédendante Y suit une loi normale $N(\mu, \sigma)$ où μ est une fonction linéaire des variables explicatives. Pour la régression logistique binomiale, la variable dépendante, aussi appelée variable réponse, suit une loi de Bernoulli de paramètre p (p étant la probabilité pour que l'événement se produise), lorsque l'expérience est répétée une fois, ou une loi Binomiale(n, p) si l'expérience est répétée n fois (par exemple la même dose est essayée sur

n patients). Dans le cas de la régression logistique, le paramètre de probabilité p est une fonction combinaison linéaire des variables explicatives X.

Le cas **binaire** est le cas où la variable réponse peut prendre 2 valeurs (correspondant à un tirage de Bernoulli), et le cas **somme de binaires** le cas où la variable réponse est le comptage du nombre de fois où l'événement d'intérêt s'est produit.

Les fonctions les plus couramment utilisées pour relier la probabilité p aux variables explicatives sont la fonction logistique (on parle alors de modèles **Logit**) et la fonction de répartition de la loi normale standard (on parle alors de modèle **Probit**). Ces deux fonctions sont parfaitement symétriques et sigmoïdes (La courbe en S représentée par la fonction $f_\lambda(x) = f(\lambda x) = \frac{1}{1+e^{-\lambda x}}$). La fonction sigmoide à la particularité d'être toujours comprise entre 0 et 1. Ainsi pour un seuil fixé entre 0 et 1 (la plupart du temps un seuil de 0.5) si le résultat de la fonction sigmoide, suivant les valeurs en entrées, est supérieur ou égale au seuil, alors on considère que l'individu est de classe 1 et sinon on considère qu'il est de classe 0. Dans le cas du dépistage de cancer du poumon on considérera ainsi que si les données d'un patient conduisent à la classe 1 alors il atteint de cancer et sinon il n'est pas atteint. A chaque résultat on attribue une probabilité qui indique son taux de certitude.

4.3.3 Les paramètres du modèle

Affichage des paramètres du modèle.

```
{'memory': None,
 'steps': [('columntransformer',
   ColumnTransformer(transformers=[('pipeline',
                                    Pipeline(steps=[('standardscaler',
                                                     StandardScaler()),
                                                    ('polynomialfeatures',
                                                     PolynomialFeatures())]),
<sklearn.compose._column_transformer.make_column_selector object at
0x000001E20BA2D580>)])),
  ('smote', SMOTE(random_state=0, sampling_strategy=0.5)),
  ('logisticregression', LogisticRegression(random_state=4))],
 'verbose': False,
 'columntransformer': ColumnTransformer(transformers=[('pipeline',
                                   Pipeline(steps=[('standardscaler',
                                                    StandardScaler()),
                                                   ('polynomialfeatures',
                                                    PolynomialFeatures())]),
<sklearn.compose._column_transformer.make_column_selector object at
0x000001E20BA2D580>)]),
 'smote': SMOTE(random_state=0, sampling_strategy=0.5),
 'logisticregression': LogisticRegression(random_state=4),
 'columntransformer__n_jobs': None,
 'columntransformer__remainder': 'drop',
 'columntransformer__sparse_threshold': 0.3,
 'columntransformer__transformer_weights': None,
 'columntransformer__transformers': [('pipeline',
```

```
   Pipeline(steps=[('standardscaler', StandardScaler()),
                   ('polynomialfeatures', PolynomialFeatures())]),
   <sklearn.compose._column_transformer.make_column_selector at
0x1e20ba2d580>)],
 'columntransformer__verbose': False,
 'columntransformer__verbose_feature_names_out': True,
 'columntransformer__pipeline': Pipeline(steps=[('standardscaler',
StandardScaler()),
                 ('polynomialfeatures', PolynomialFeatures())]),
 'columntransformer__pipeline__memory': None,
 'columntransformer__pipeline__steps': [('standardscaler', StandardScaler()),
  ('polynomialfeatures', PolynomialFeatures())],
 'columntransformer__pipeline__verbose': False,
 'columntransformer__pipeline__standardscaler': StandardScaler(),
 'columntransformer__pipeline__polynomialfeatures': PolynomialFeatures(),
 'columntransformer__pipeline__standardscaler__copy': True,
 'columntransformer__pipeline__standardscaler__with_mean': True,
 'columntransformer__pipeline__standardscaler__with_std': True,
 'columntransformer__pipeline__polynomialfeatures__degree': 2,
 'columntransformer__pipeline__polynomialfeatures__include_bias': True,
 'columntransformer__pipeline__polynomialfeatures__interaction_only': False,
 'columntransformer__pipeline__polynomialfeatures__order': 'C',
 'smote__k_neighbors': 5,
 'smote__n_jobs': None,
 'smote__random_state': 0,
 'smote__sampling_strategy': 0.5,
 'logisticregression__C': 1.0,
 'logisticregression__class_weight': None,
 'logisticregression__dual': False,
 'logisticregression__fit_intercept': True,
 'logisticregression__intercept_scaling': 1,
 'logisticregression__l1_ratio': None,
 'logisticregression__max_iter': 100,
 'logisticregression__multi_class': 'auto',
 'logisticregression__n_jobs': None,
 'logisticregression__penalty': 'l2',
 'logisticregression__random_state': 4,
 'logisticregression__solver': 'lbfgs',
 'logisticregression__tol': 0.0001,
 'logisticregression__verbose': 0,
 'logisticregression__warm_start': False}
```

- **penalty (pénalité)** : Ce paramètre est utilisé pour spécifier la norme (L1 ou L2) utilisée dans la pénalisation (régularisation). Il peut prendre les valeurs : 'L1', 'L2', 'elasticnet' ou none (aucun).
- **dual (double)** : Il est utilisé pour la formulation double ou primale. La formulation double n'est mise en œuvre que pour la pénalité L2. Il prend des valeurs booléennes.

- **tol (tolérance)** : Il représente la tolérance pour les critères d'arrêt.
- **C** : Il représente l'inverse de la force de régularisation. Il doit toujours être une décimale positive.
- **fit_intercept** : Ce paramètre spécifie si une constante (biais ou interception) doit être ajoutée à la fonction de décision. Il prend des valeurs booléennes.
- **intercept_scaling** : Ce paramètre est utile lorsque le solveur 'liblinear' est utilisé ou que 'fit_intercept' est défini sur vraie. Il prend des valeurs décimales.
- **class_weight** : Il représente les poids associés aux classes. Si nous utilisons l'option par défaut ('none' qui veut dire aucun), cela signifie que toutes les classes sont censées avoir un poids égal à 1. D'autre part, si vous choisissez la valeur 'équilibré', il utilisera les valeurs de y pour ajuster automatiquement les poids.
- **random_state** : Ce paramètre représente la graine du nombre pseudo aléatoire généré qui est utilisé lors du brassage des données. Voici les options -
  - **int (entier)** : dans ce cas, random_state est la graine utilisée par le générateur de nombres aléatoires.
  - **'Instance RandomState'** : dans ce cas, random_state est le générateur de nombres aléatoires.
  - **Aucun** : dans ce cas, le générateur de nombres aléatoires est l'instance RandonState utilisée par np.random.
- **solver (solveur)** : Ce paramètre représente l'algorithme à utiliser dans le problème d'optimisation. Voici les propriétés des options sous ce paramètre -
  - **liblinear** : C'est un bon choix pour les petits ensembles de données. Il gère également la pénalité L1. Pour les problèmes multi classes, il est limité aux schémas à un contre repos.
  - **newton-cg** : Il ne gère que la pénalité L2.
  - **lbfgs** : Pour les problèmes multi classes, il gère la perte multinomiale. Il ne gère également que la pénalité L2.
  - **saga** : C'est un bon choix pour les grands ensembles de données. Pour les problèmes multiclasses, il gère également la perte multinomiale. En plus de la pénalité L1, il prend également en charge la pénalité « elasticnet ».
  - **sag** : Il est également utilisé pour les grands ensembles de données. Pour les problèmes multi classes, il gère également la perte multinomiale.
- **max_iter** : Comme son nom l'indique, il représente le nombre maximal d'itérations prises pour que les solveurs convergent.
- **multi_class** :
  - **ovr** : Pour cette option, un problème binaire convient à chaque étiquette.
  - **multimonial** : Pour cette option, la perte minimisée est l'ajustement de la perte multinomiale sur l'ensemble de la distribution de probabilité. Nous ne pouvons pas utiliser cette option si solver = 'liblinear'.
  - **auto** : Cette option sélectionnera 'ovr' si solver = 'liblinear' ou si les données sont binaires, sinon il choisira 'multinomial'.

- **verbose** : Par défaut, la valeur de ce paramètre est 0, mais pour le solveur liblinear et lbfgs, nous devons définir verbose sur n'importe quel nombre positif.
- **warm_start** : Avec ce paramètre défini sur True, nous pouvons réutiliser la solution de l'appel précédent pour l'adapter en tant qu'initialisation. Si nous choisissons la valeur par défaut, c'est-à-dire faux, cela effacera la solution précédente.
- **n_jobs** : Si multi_class = 'ovr', ce paramètre représente le nombre de cœurs de CPU utilisés lors de la parallélisation sur les classes. Il est ignoré lorsque solver = 'liblinear'.
- **l1_ratio** : Il est utilisé dans le cas où penalty = 'elasticnet'. Il s'agit essentiellement du paramètre de mélange Elastic-Net avec 0 < = l1_ratio > = 1.

Définition des principaux termes:

- Régularisation : À la base, la régularisation tente de limiter le surapprentissage. La régularisation peut s'introduire dans la fonction de coût (fonction de perte ou d'erreurs que l'on doit minimiser). Les normes L1 ou L2 sont des termes ajouté à la fonction de coût en tant que terme de régularisation. L'ajout de tels termes de régularisation à la fonction de coût est un concept très populaire en apprentissage machine. θ_i représente l'estimation (ou valeur prédite) d'un paramètre du modèle.
- Norme *L1* : $\lambda \sum_{i=1}^{n} |\theta_i|$. La régularisation par norme L1 (Lasso) tente de minimiser la somme des différences absolues entre valeurs réelles et valeurs prédites. Linéaire, elle offre la possibilité au modèle de facilement fixer un poids à 0 et peut donc, entre autres, faciliter la sélection de caractéristiques en forçant une représentation éparse.
- Norme *L2* : $\lambda \sum_{i=1}^{n} \theta_i^2$. La régularisation par norme L2 (Ridge / Tikhonov) tente de minimiser la somme des carrées des différences entre valeurs réelles et valeurs prédites. Ce terme est, entre autres, plus rapide à calculer que le terme L1. Exponentielle, elle promouvoit plutôt une représentation diffuse et, de ce fait, performe généralement mieux que la L1.
- Poids de régularisation (hyperparamètre) : l'ampleur de l'effet du terme de régularisation est contrôlé grâce à un poids λ placé à l'avant du terme.

4.4 Optimisation de la performance du modèle

Optimisons la performance du modèle avec une validation croisée stratifiée et la recherche par grilles aléatoires. L'un nous permettra d'effectuer des validations croisées en sélectionnant les ensembles par strates (par rapport à chaque modalité) et l'autre utilisera le premier pour vérifier la combinaison des paramètres du modèle qui donnent le meilleur score. On prendra comme métrique la sensibilité pour la recherche des meilleurs paramètres du modèle.

Il est important de noter que la recherche par grille ne sera pas effectuée sur toutes les valeurs des paramètres mais se fera plutôt par sélections aléatoires.

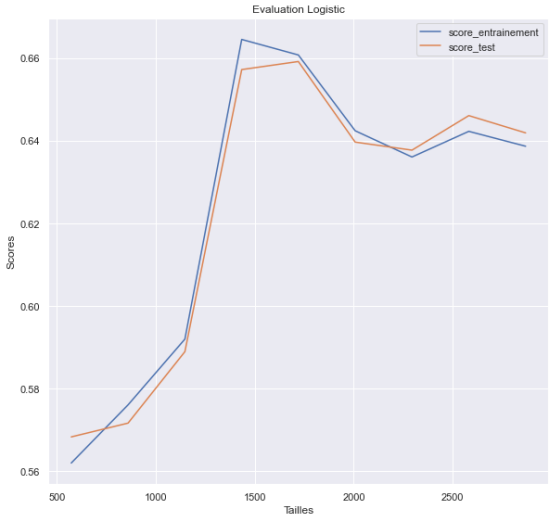
Conclusion : Nous avons choisi un nombre de valeurs très important pour chaque paramètre en tenant compte des disparités existantes entre elles. Nous évaluons le temps d'exécution de l'algorithme (de la recherche par grilles jusqu'à l'évaluation du meilleur modèle) entre 1 et 2 minutes sur le support utilisé pour les tests. Cependant sur des appareils beaucoup moins adaptés à ce genre de calculs nous pouvons noter un temps d'exécution pouvant excéder les 2 minutes.

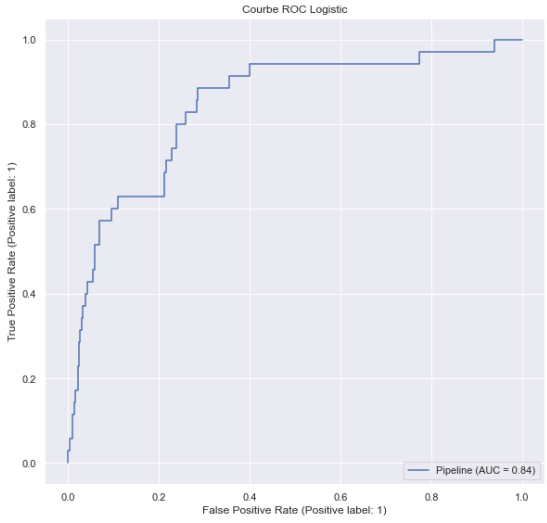
Les meilleurs paramètres choisis avec une graine égale à 19.

```
{'smote__sampling_strategy': 0.9,
 'smote__k_neighbors': 9,
 'logisticregression__solver': 'saga',
 'logisticregression__penalty': 'l2',
 'logisticregression__fit_intercept': True,
 'logisticregression__dual': False,
 'logisticregression__C': 5.3,
 'columntransformer__pipeline__polynomialfeatures__degree': 11}
```

Résultats obtenus avec le meilleur modèle.

```
La precision du modèle : 0.6688963210702341
Matrice de confusion :
[[569 293]
 [  4  31]]
Rapport de classification :
              precision    recall  f1-score   support

           0       0.99      0.66      0.79       862
           1       0.10      0.89      0.17        35

    accuracy                           0.67       897
   macro avg       0.54      0.77      0.48       897
weighted avg       0.96      0.67      0.77       897
```

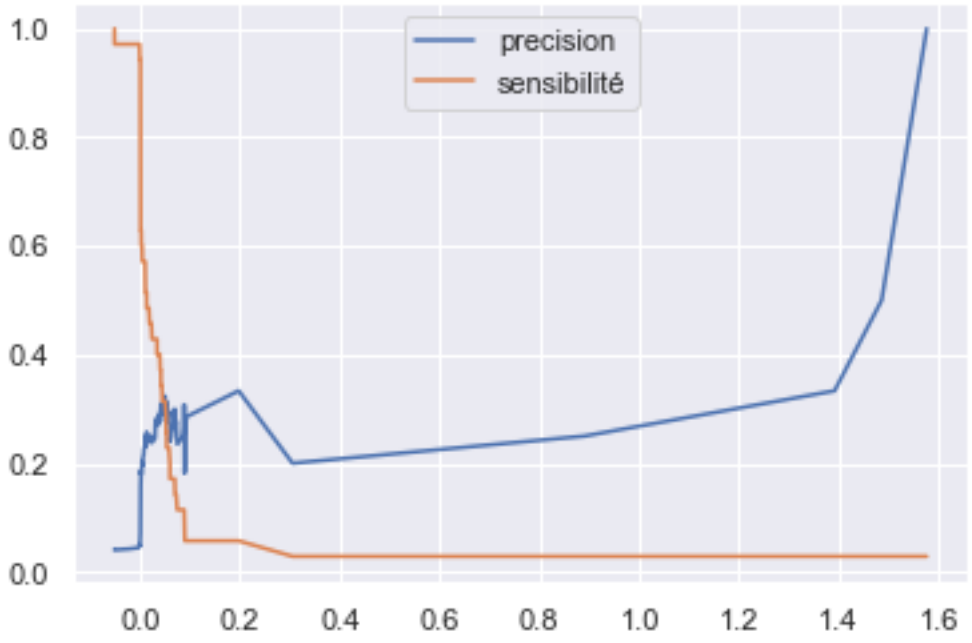




Avec les paramètres optimaux choisis aléatoirement nous obtenons une sensibilité de **89%** ($> 80\%$) et une aire sous la courbe de **84%** ($>80\%$) et nous n'avons quasiment plus de sur entraînement observable. Donc nous pouvons dire que nous avons atteint l'objectif fixé.

4.5 Compromis entre Sensibilité et Précision

Tracons la courbe de Sensibilité-Précision pour vérifier quel seuil nous permet d'obtenir une bonne sensibilité sans pour autant trop diminuer la précision.



Calcul de la sensibilité

Nous obtenons ainsi une sensibilité finale de **94%** pour un seuil de -0.001

4.6 Conclusion

- La phase d'exploration des données nous a permis de détecter des anomalies, notamment les éléments suivants :
  - Déséquilibre de classes (variable stroke) ;
  - Valeurs manquantes dans la variable bmi ;
  - Valeurs aberrantes dans les variables quantitatives ;
  - Valeurs inconsistantes dans la variable smoking_type ;
  - Corrélations entre variables ;
  - Variables inutiles dans l'analyse (variable id) ;
  - Variable n'apportant pas d'informations.
- A travers une démarche **évaluation -> idée -> code**, on a déterminé comment traiter idéalement les données pour obtenir une amélioration du modèle de test (l'arbre de décision). On obtient un plus bon score sur le modèle de test en effectuant les prétraitements suivant :

- Suppression de variables jugées inutiles ou fortement corrélées avec la variable age ou avg_glucose_level ; - Suppression de valeurs aberrantes pour obtenir une meilleure *auc* ; - Séparation des données en données d'entraînement et de test ; - Sélection de variables après première

évaluation ; - Sur échantillonnage des données après deuxième évaluation.

- Entraînement de plusieurs modèles (avec des données quantitatives discrétisées et sur échantillonnage de la classe minoritaire) et choix du meilleur modèle qui est la régression logistique ;
- Optimisation du modèle choisi en choisissant les paramètres lui fournissant le meilleur score (meilleure sensibilité et meilleure auc) ;
- On obtient une sensibilité de 89% et une sensibilité de 84% après la recherche par grille ;
- On essaie de trouver un seuil optimal pour la sensibilité (pour un seuil de -0.001 on obtient une sensibilité de 94%).

4.7 Procédure de déploiement du modèle

Le déploiement du modèle va nécessiter la librairie joblib qui va nous permettre de sauvegarder le modèle et la conception d'un package et d'un environnement virtuel à l'aide de la librairie poetry afin d'exécuter des tests et des prédictions après chargement du modèle retenu. La méthode argparse permet d'envoyer directement les valeurs des variables age et taux de glucose sous forme de paramètres à un fichier predict.py qui va donner directement le résultat souhaité. Cependant étant donné le manque de complétude dans la collecte de données il est plus judicieux de ne pas utiliser ce modèle pour la détection d'AVC (*consulter un médecin serait plus judicieux pour le moment*).