stroke-prediction-article

February 11, 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 ceux qui sont vraiment malades parmi ceux qui sont identifiés comme malade). Sinon un patient malade peut être diagnostiquer comme sain alors que ce n'est pas le cas. Nous préférerons ainsi la sensibilité/spécificité 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/spécificité supérieure à **80%**. Nous pouvons également utiliser des outils supplémentaires comme **les courbes ROC et Precision-Recall** pour affiner nos choix de performance. ## EXPLORATION DES DONNEES ### ANALYSE DE LA FORME **CIBLE** : La variable dépendante est la variable **stroke** qui contient des données discrètes. 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, 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 (Un peu entassées au niveau des premières observations). Mais en sachant que le nombre de valeurs manquantes ne représente que 16% dans 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 dans la variable bmi constitue une bonne stratégie.

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

ETUDE PLUS POUSSEE DES VARIABLES AVEC PANDAS-PROFILING ##

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 corréla-

tions 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 coincident 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

Vérifions le contenu des dix premières lignes

[11]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1
5	56669	Male	81.0	0	0	Yes	Private	Urban	186.21	29.0	formerly smoked	1
6	53882	Male	74.0	1	1	Yes	Private	Rural	70.09	27.4	never smoked	1
7	10434	Female	69.0	0	0	No	Private	Urban	94.39	22.8	never smoked	1
8	27419	Female	59.0	0	0	Yes	Private	Rural	76.15	NaN	Unknown	1
9	60491	Female	78.0	0	0	Yes	Private	Urban	58.57	24.2	Unknown	1

```
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.

Déterminons le type de la cible

[12]:
```
dtype('int64')
```

Vérifions les valeurs possibles du target

[13]:
```
array([1, 0], dtype=int64)
```

2.2 Nombre de ligne et de colonnes

[14]:
```
(5110, 12)
```

Nous avons 5110 observations et 12 variables (dont la cible)

2.3 Types des variables explicatives

Identifions les types des variables

[15]:
```
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.

Pour chaque variable de type object vérifions ses catégories

```
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']
```

Vérifions les valeurs que contiennent les variables hypertension et heart_disease

```
Unique values of hypertension-------- [0 1]

Unique values of heart_disease------- [1 0]
```

Vérifions les statistiques des colonnes non catégorielles

[20]:

	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.

[21]:

```
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).

Nous remarquons que seule la variable bmi contient des valeurs manquantes qui sont un peu dispersées dans le dataset (Un peu entassées au niveau des premières observations).

Vérifions le pourcentage de valeurs manquantes.

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

2.5 Identification des observations redondantes

[26]:
```
0
```

Le dataset ne contient pas d'observations dupliquées.

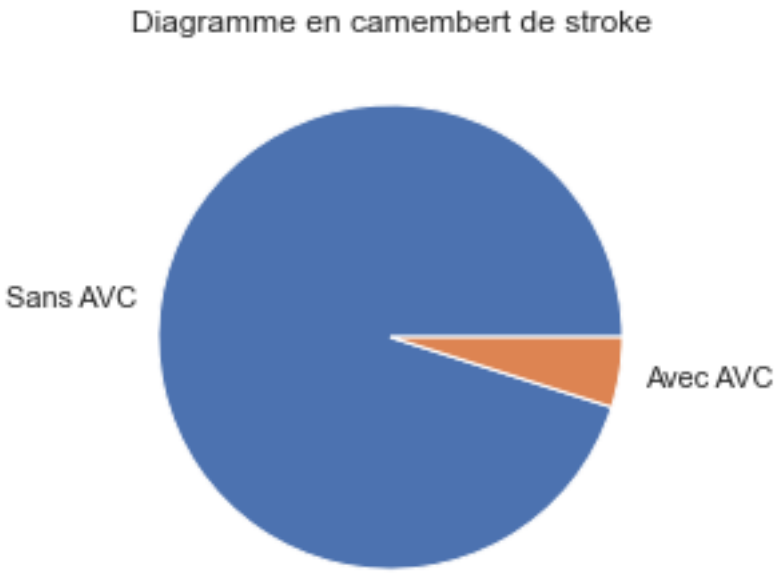
2.6 Visualisation de la cible

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

[27]:
```
0    95.127202
1     4.872798
Name: stroke, dtype: float64
```

Visualisons la proportion de chaque classe à l'aide d'un diagramme en camembert.

[28]:
```
Text(0, 0.5, '')
```



Nous remarquons que 95.13% des personnes sont atteintes d'AVC contre seulement 4.87% de personnes atteintes d'AVC.

2.7 Compréhension des différentes variables

Analysons les variables à l'aide de graphiques.

[31]:
```
hypertension      category
heart_disease     category
gender            category
ever_married      category
work_type         category
Residence_type    category
smoking_status    category
dtype: object
```

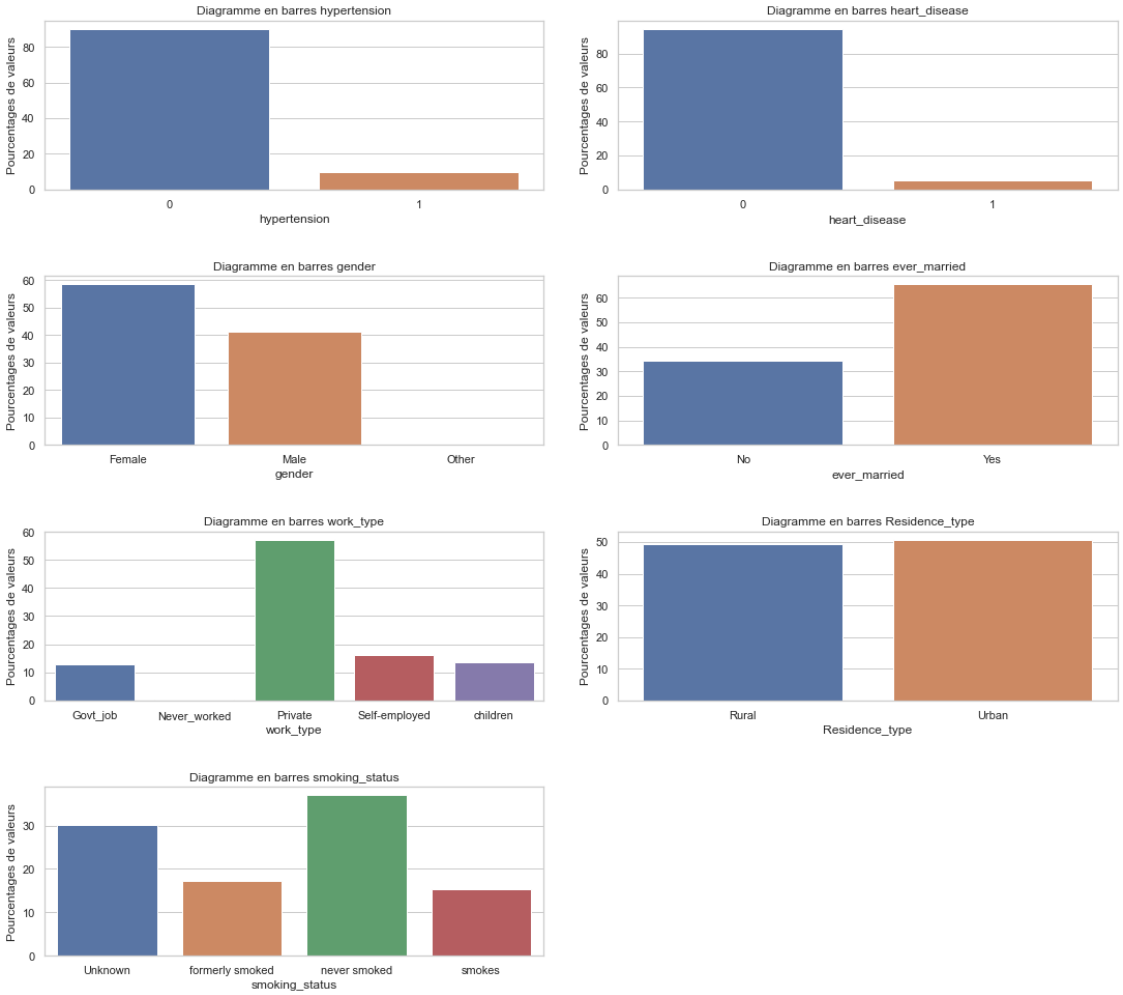
2.7.1 Variables catégorielles

Identifions la proportion de chaque modalité au sein de sa classe.

```
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
--------------------
```

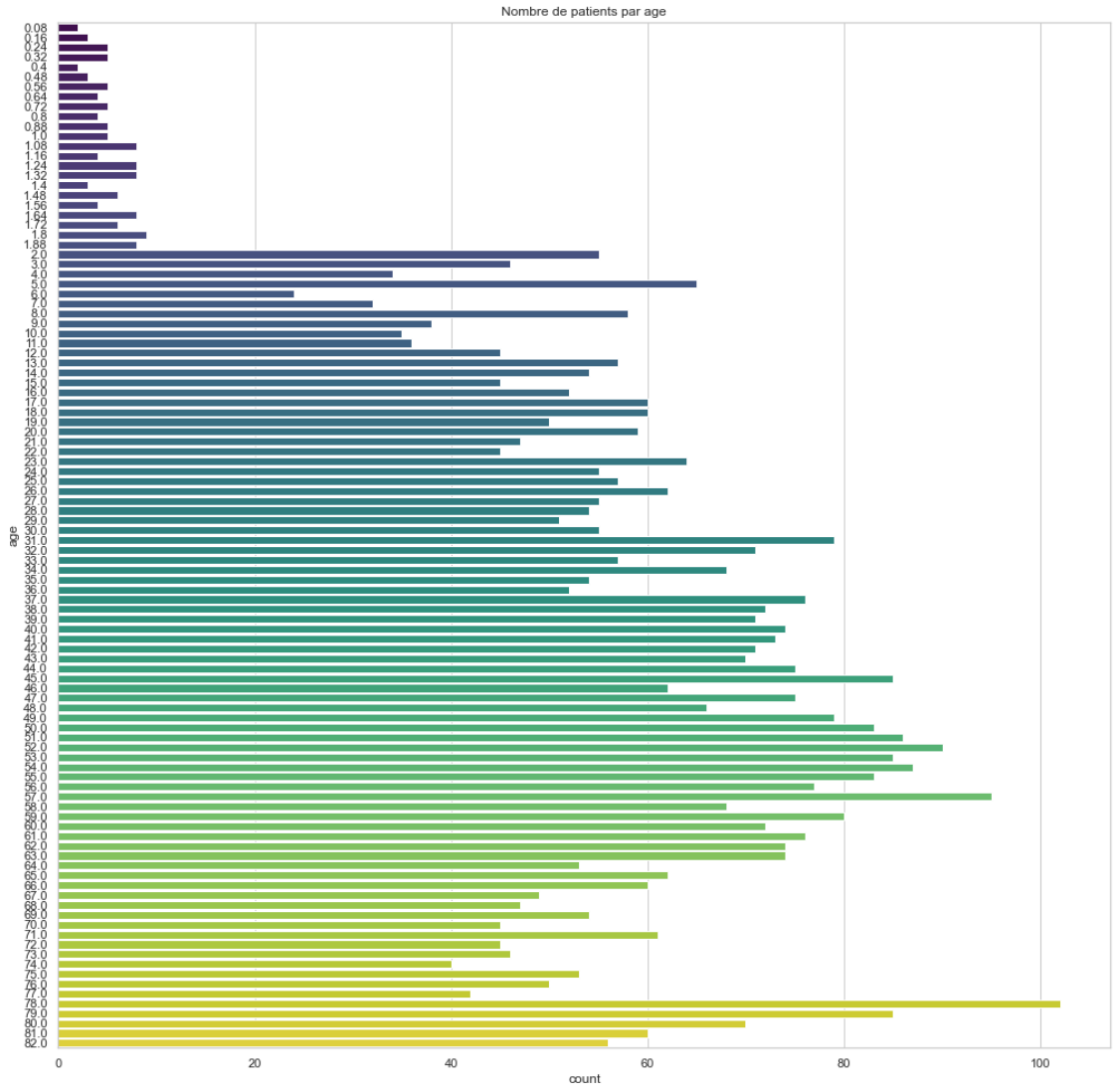
Tracons le diagramme en barres des proportions pour chaque variable



2.7.2 Variables non catégorielles

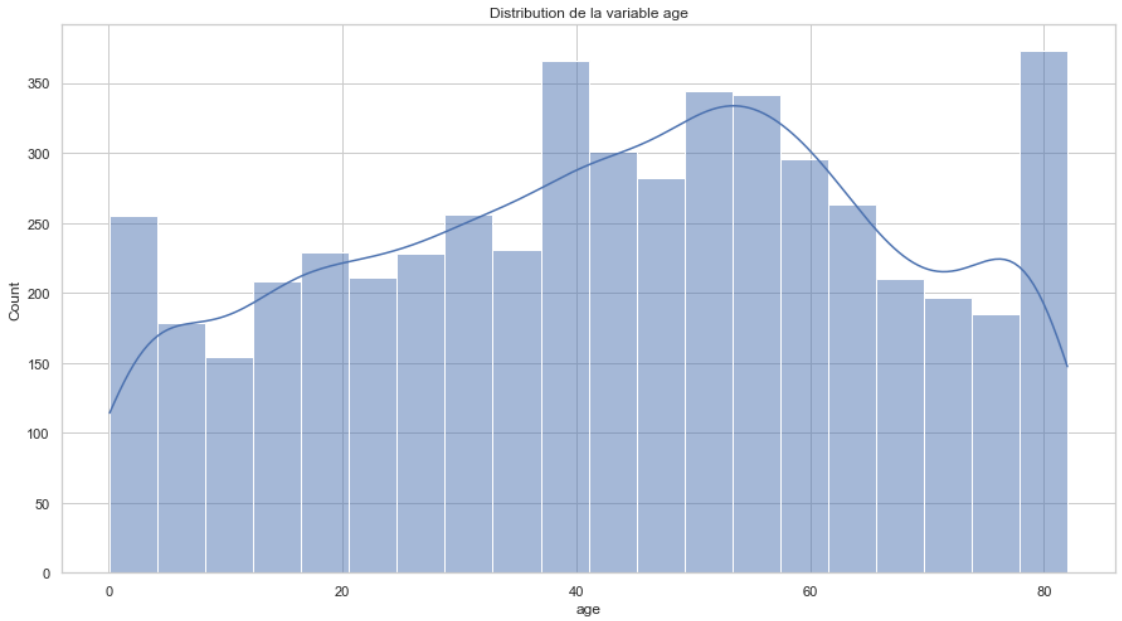
Comptons le nombre de patients par age à l'aide d'un countplot.

[34]: `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. Visualisons la distribution de la variable age.

[35]: `Text(0.5, 1.0, 'Distribution de la variable age')`

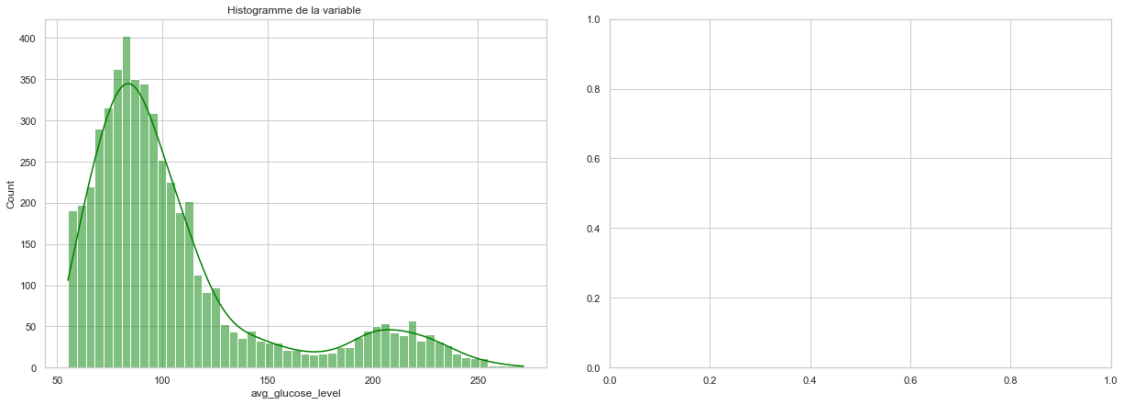


La variable age ne suit pas une distribution normale. Nous remarquons que la distribution est plus élevée au niveau des patients âgés entre 37 et 63 ans. Elle est moins élevée au niveau des jeunes patients.

Traçons les histogrammes du niveau moyen de glucoses dans le sang et celui de l'indice de masse corporelle.

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-36-706d6be17674> in <module>
      7     color = "green" if i == 0 else "blue"
      8     sns.histplot(data = data_frame, x = column, kde = True, ax = axs[i]␣
 ↪color = color)
----> 9     axs[i].set_title("Histogramme de la variable ", column)
     10

C:\anaconda3\lib\site-packages\matplotlib\axes\_axes.py in set_title(self,␣
 ↪label, fontdict, loc, pad, y, **kwargs)
    166         title.update(default)
    167         if fontdict is not None:
--> 168             title.update(fontdict)
    169         title.update(kwargs)
    170         return title

C:\anaconda3\lib\site-packages\matplotlib\text.py in update(self, kwargs)
    162     def update(self, kwargs):
    163         # docstring inherited
--> 164         kwargs = cbook.normalize_kwargs(kwargs, Text)
```

```
    165         sentinel = object()  # bbox can be None, so use another sentinel.
    166         # Update fontproperties first, as it has lowest priority.

C:\anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in␣
 ↪normalize_kwargs(kw, alias_mapping)
   1734     ret = {}  # output dictionary
   1735
-> 1736     for k, v in kw.items():
   1737         canonical = to_canonical.get(k, k)
   1738         if canonical in canonical_to_seen:

AttributeError: 'str' object has no attribute 'items'
```



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 croisés (de comptage des modalités).

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.

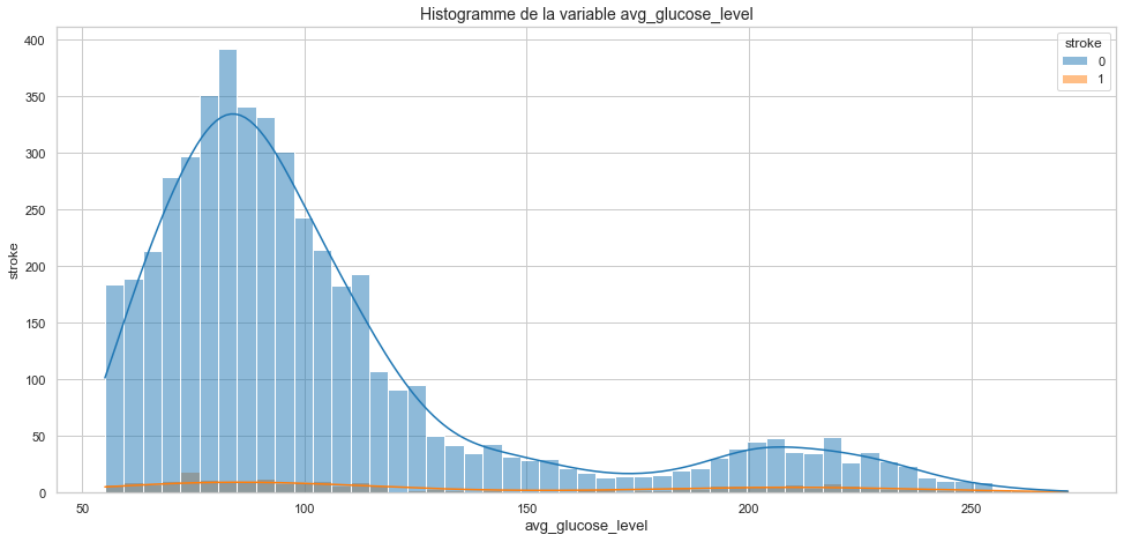
2.8.2 Variables non catégorielles / variable cible

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

Pour la variable age

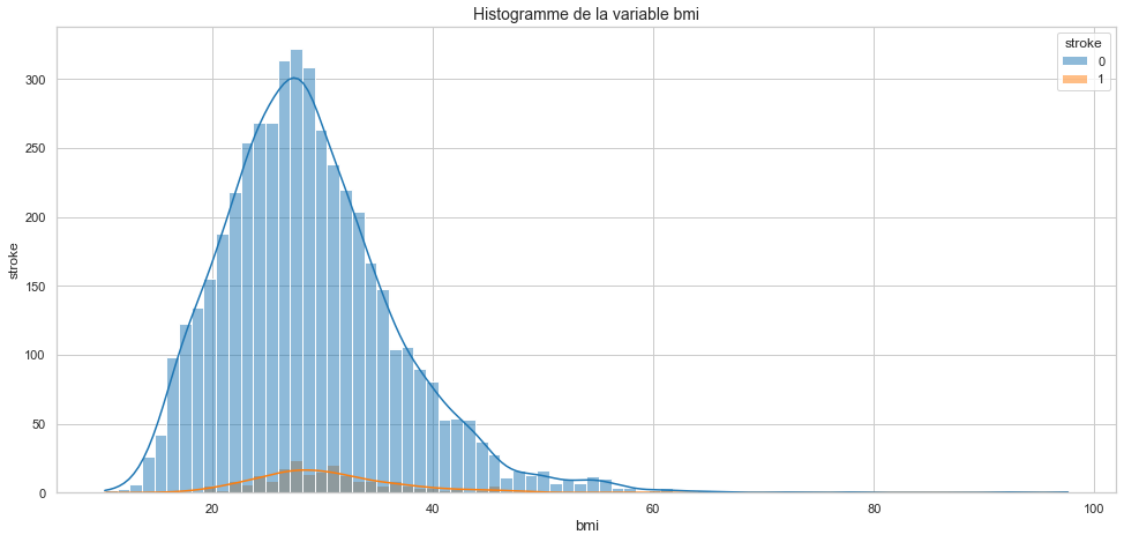
Nous voyons que les deux distributions sont différentes. Nous constatons que les personnes plus agées ont plus de risque d'être atteintes d'AVC.

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

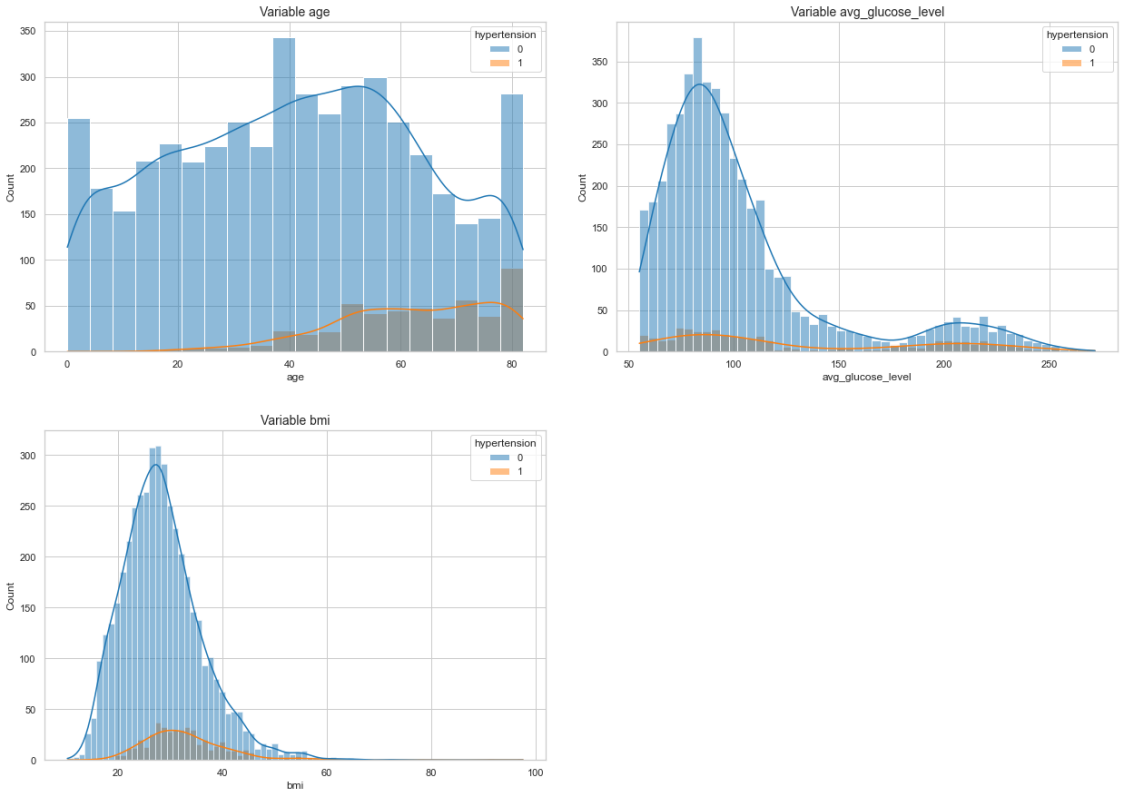
Vérifions si certaines variables non catégorielles ont une forte corrélation entre elles.

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

2.10 Relations entre variables catégorielles et non catégorielles

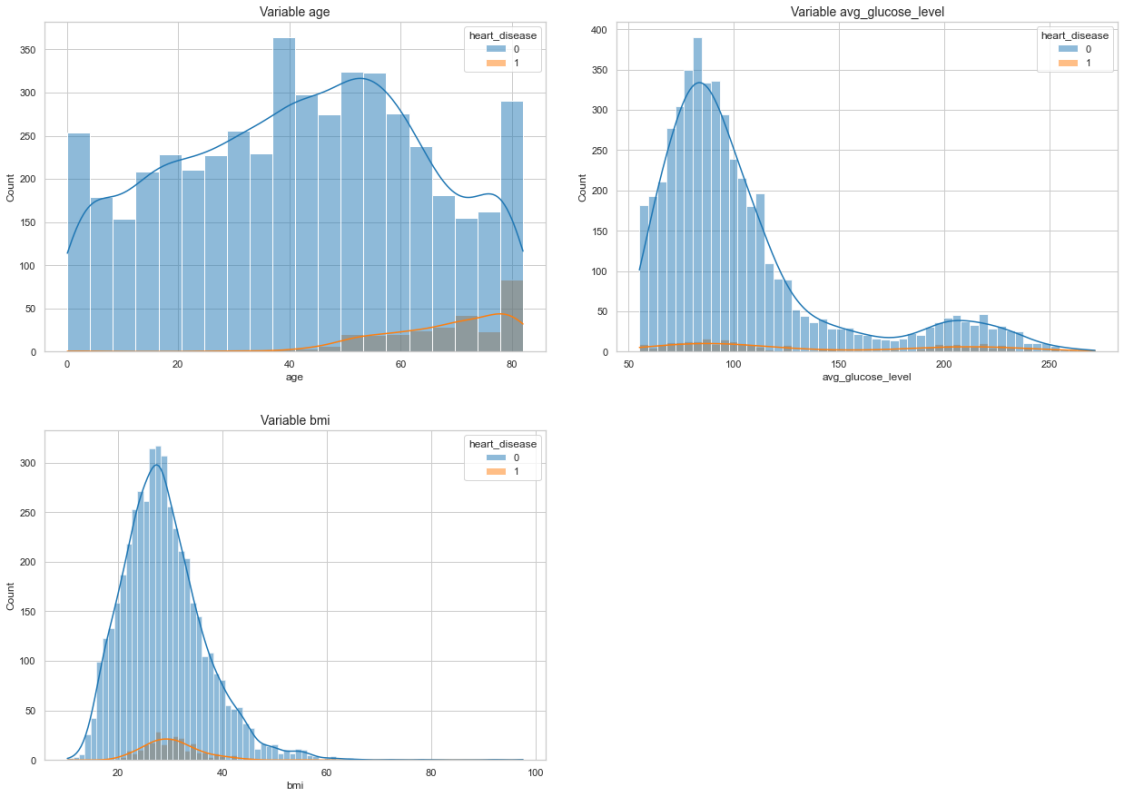
Nous allons vérifier pour chaque variable quantitative si elle est influencée par une ou plusieurs variable(s) catégorielle(s).

Pour la variable hypertension



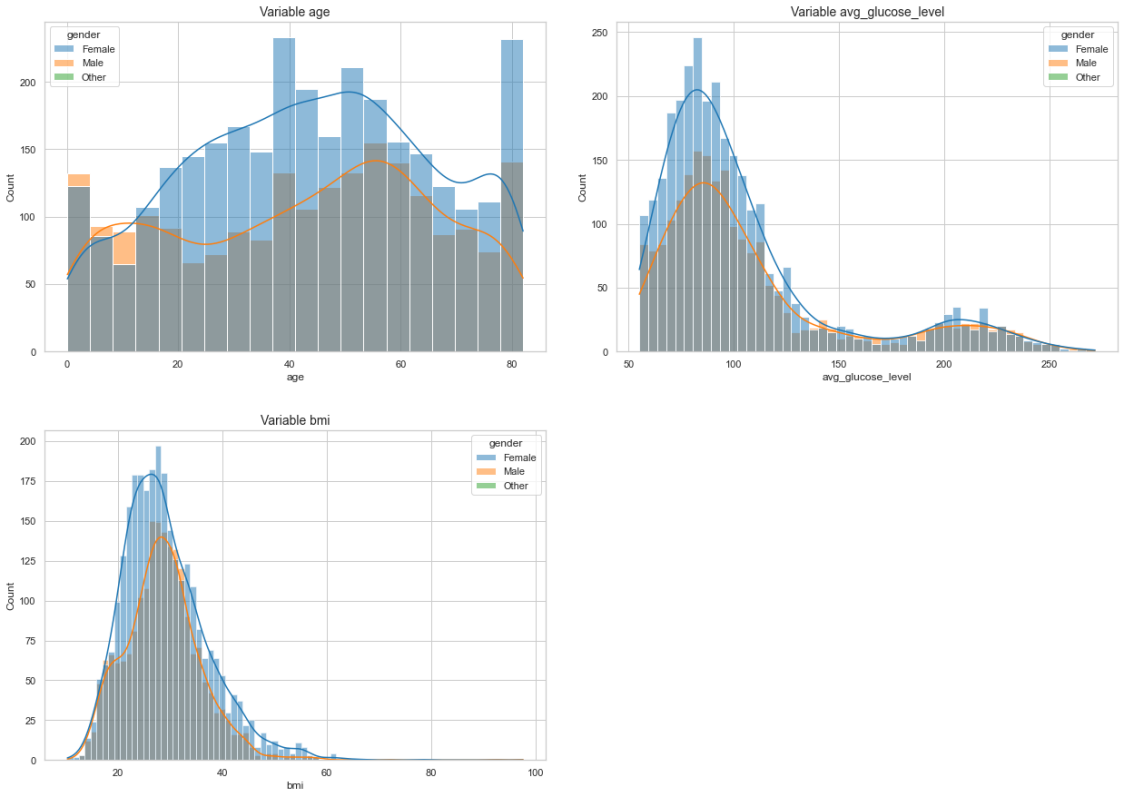
Nous remarquons une légère dépendance entre les variables hypertension et age.

Pour la variable heart_disease



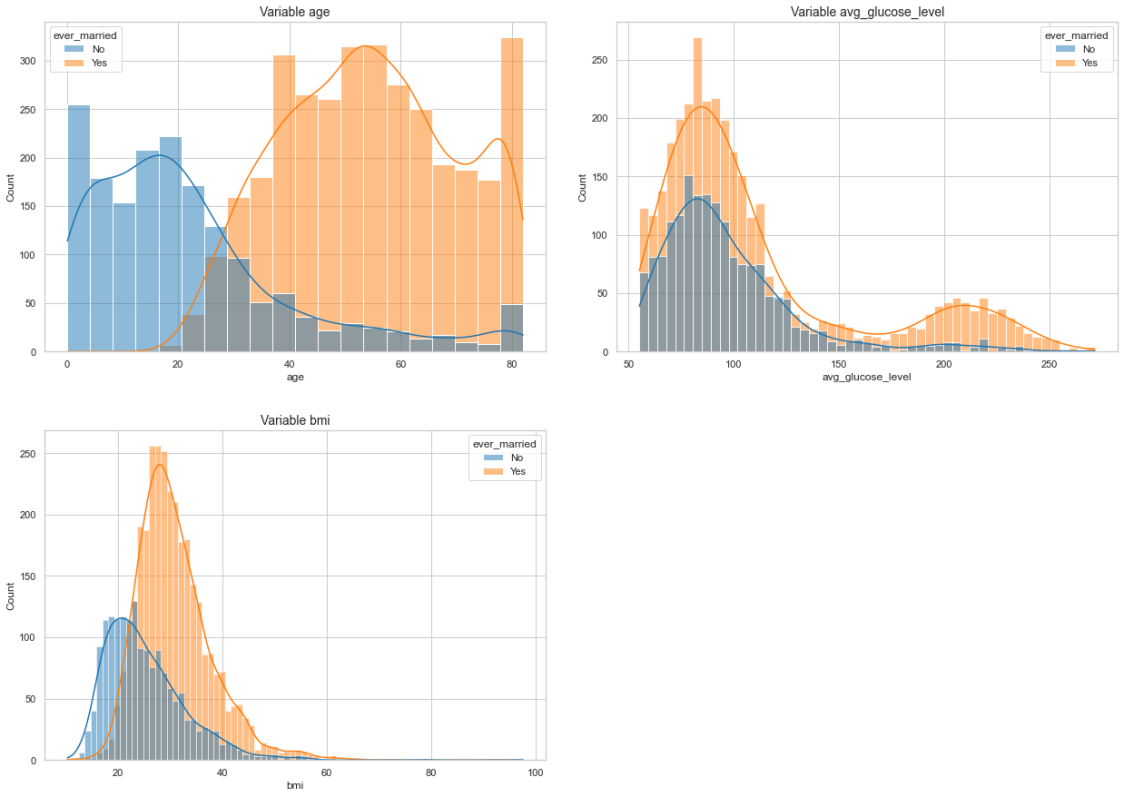
Nous constatons également une légère dépendance entre les variables heart_disease et age.

Pour la variable gender



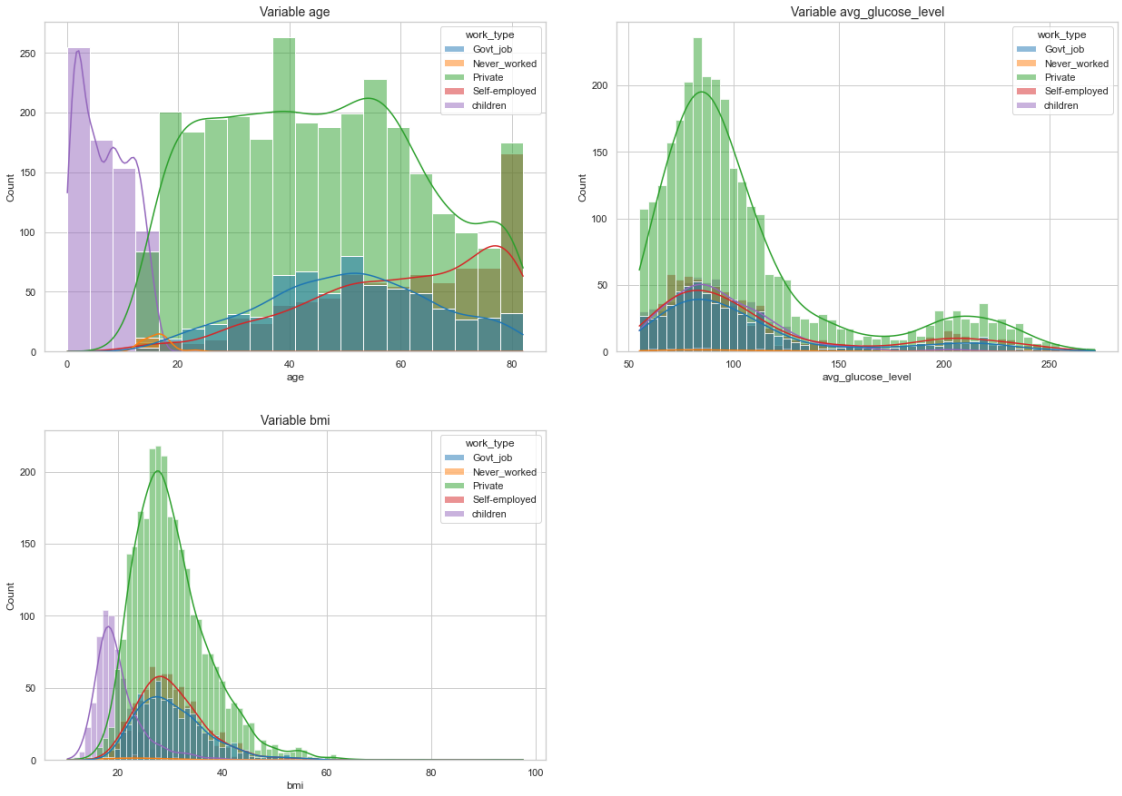
Nous ne remarquons aucune corrélation entre la variable age et gender

Pour la variable ever_married



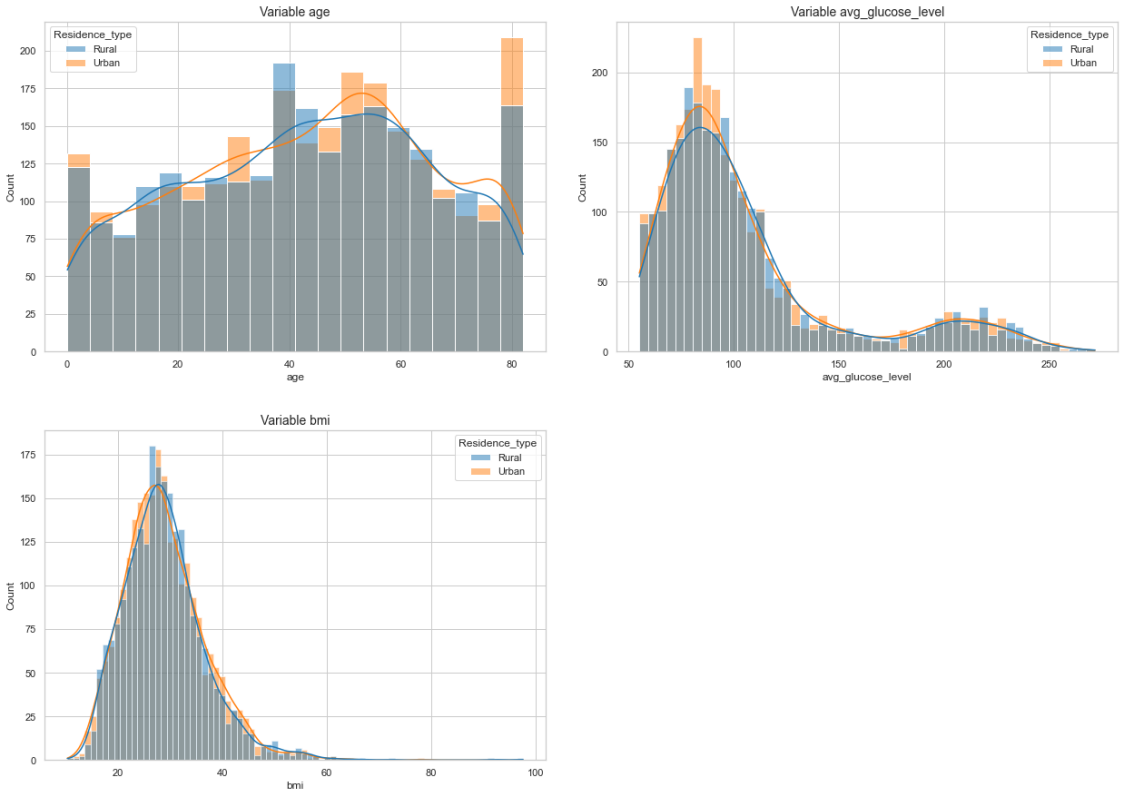
La variable ever_married est fortement corrélée avec les variables age et bmi.

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).

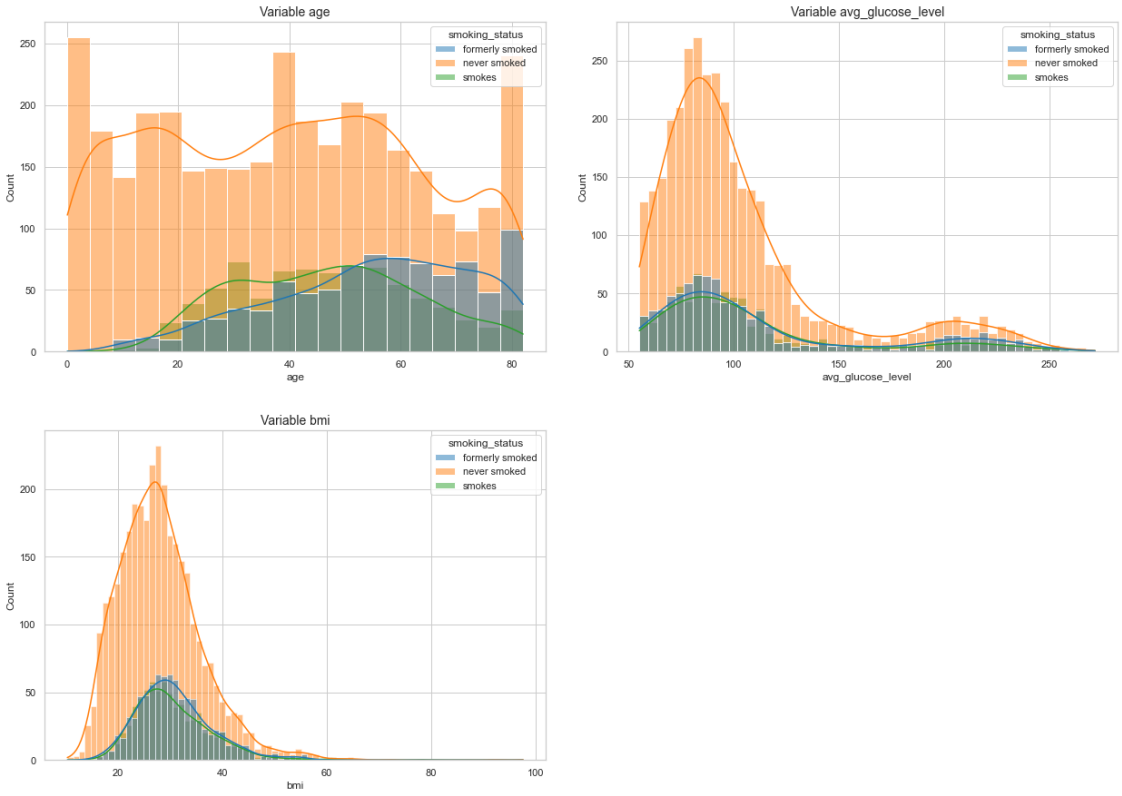
Pour la variable residence_type



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.

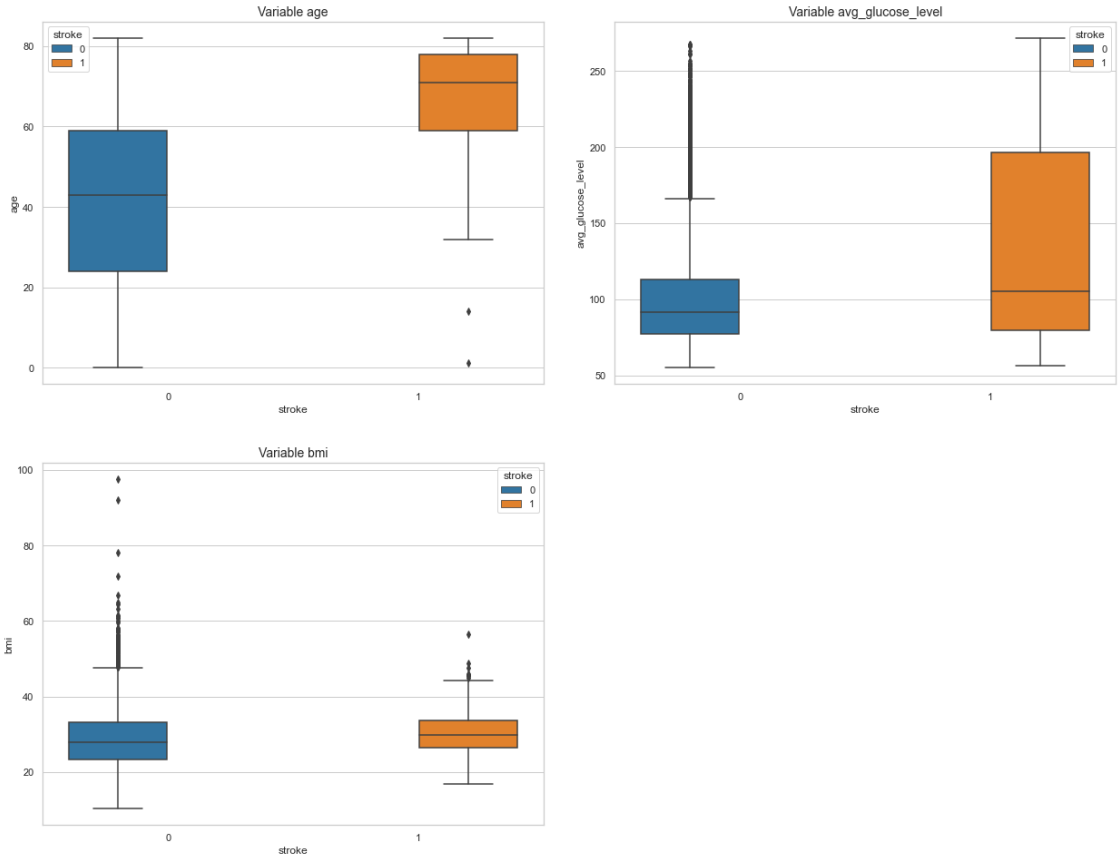
[49]:
```
never smoked       67.240705
formerly smoked    17.318982
smokes             15.440313
Name: smoking_status, dtype: float64
```



La variable smoking_status est fortement corrélée avec la variable age.

2.11 Identification des valeurs aberrantes

Tracons les boxplots des variables catégorielles en les séparant par classe de patients (atteints d'AVC ou pas).



Nous constatons que toutes 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 avec un test de student.

Nous devons d'abord séparer le dataframe entre les catégories test positif et test negatif

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

Selon le test seule la variable bmi n'influence pas le fait qu'un patient soit malade ou pas.

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.
- **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**. 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 plus bon score pour la classe 1 qui était considérée comme un bruit à cause du déséquilibre de classe. Ainsi, il sera maintenant soutenu par les valeurs 'anormales' de la variable avg_glucose_level. 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** :
- **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 négatif* et seulement **20%** pour la class *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 :
- **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.
- **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.

3.2 suppression de variables

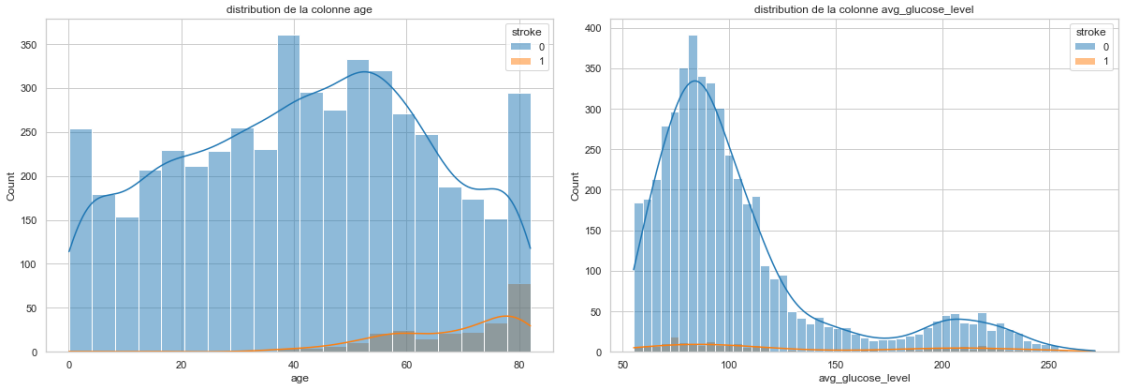
Supprimons les colonnes bmi, work_type, ever_married, smoking_status et id.

Vérifions quelles sont les variables restantes

[58]:
```
Index(['gender', 'age', 'hypertension', 'heart_disease', 'Residence_type',
       'avg_glucose_level', 'stroke'],
      dtype='object')
```

3.3 Suppression des données aberrantes

Traçons les distributions des deux variables quantitatives en cas de suppression de données aberrantes.



Vérifions combien d'observations il nous reste.

[61]:
```
5110
```

3.4 Encodage des données qualitatives

Nous devons récupérer dans une nouvelle variable le nouveau dataframe dont les données qualitatives seront encodées.

3.5 Séparation des données

Séparons les données d'entraînement des données de test.

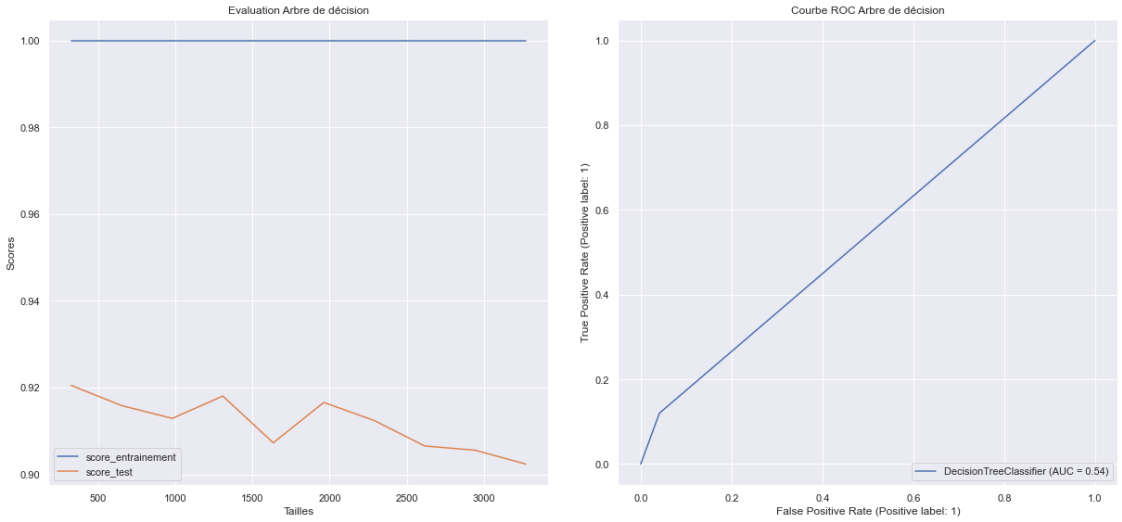
Vérifions les tailles des variables explicatives

```
Taille des données d'entraînement :  4088  observations
Taille des données de test :  1022  observations
```

3.6 Premier entraînement avec un arbre de décision

```
La precision du modèle : 0.9178082191780822
Matrice de confusion :
```

```
[[932  40]
 [ 44   6]]
Rapport de classification :
              precision    recall  f1-score   support

           0       0.95      0.96      0.96       972
           1       0.13      0.12      0.12        50

    accuracy                           0.92      1022
   macro avg       0.54      0.54      0.54      1022
weighted avg       0.91      0.92      0.92      1022
```



On obtient une aire sous la courbe de **54%**.

3.7 Sélection de variables

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

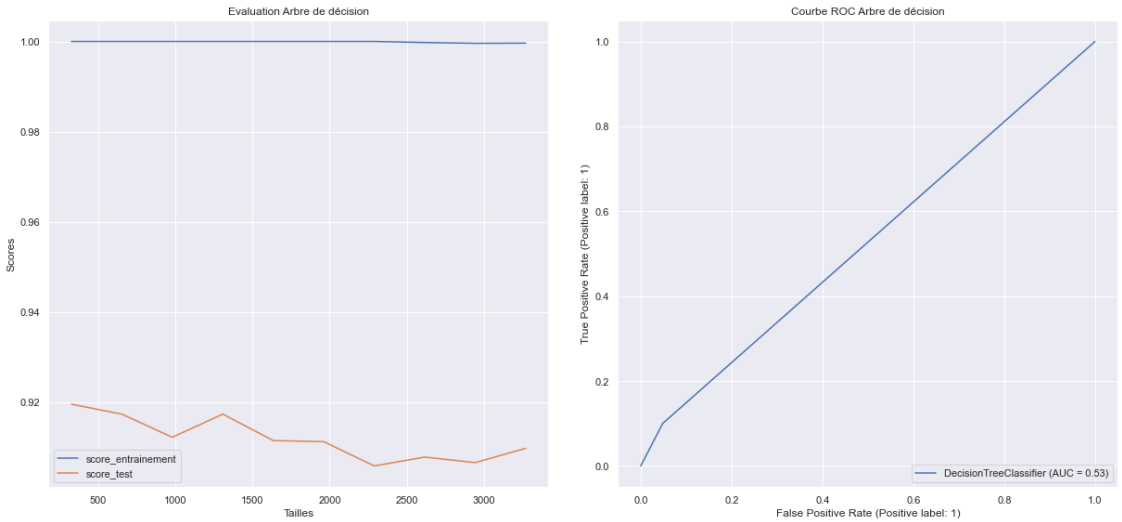
```
Colonnes choisies :
 Index(['age', 'avg_glucose_level'], dtype='object')
```

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

```
La precision du modèle : 0.9099804305283757
Matrice de confusion :
[[925  47]
 [ 45   5]]
Rapport de classification :
```

	precision	recall	f1-score	support
0	0.95	0.95	0.95	972
1	0.10	0.10	0.10	50
accuracy			0.91	1022
macro avg	0.52	0.53	0.53	1022
weighted avg	0.91	0.91	0.91	1022

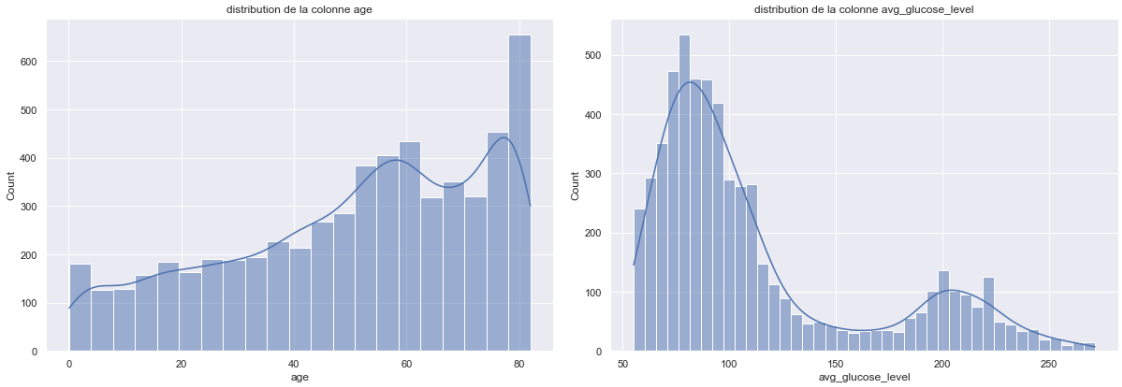


Nous notons une amélioration de la sensibilité.

3.9 Sur échantillonnage des données

```
Comptage des classes avant sur échantillonnage :
Counter({0: 3889, 1: 199})
Comptage des classes après sur échantillonnage :
Counter({0: 3889, 1: 1944})
```

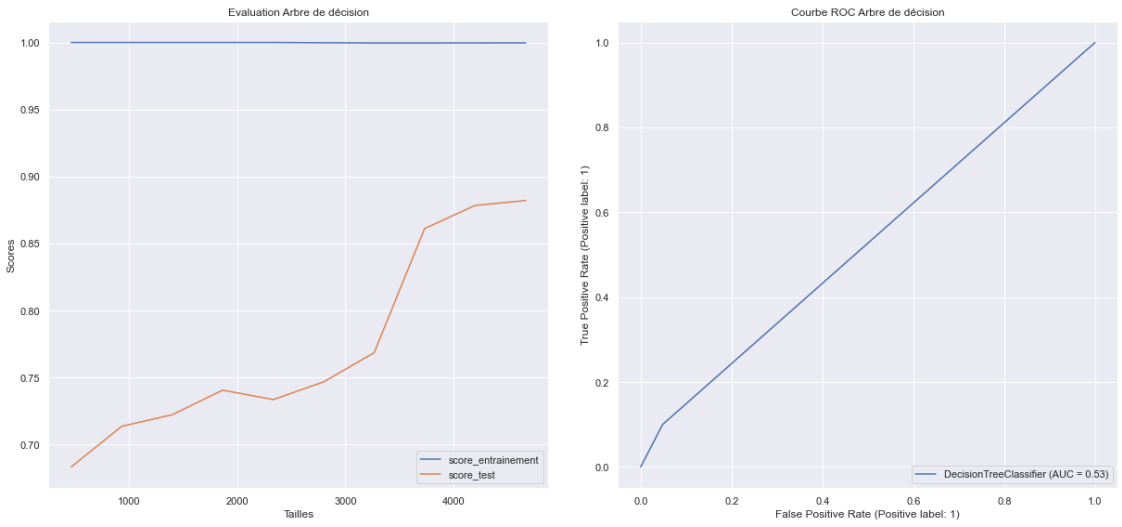
Tracons les distributions des variables age et avg_glucose_level obtenues après sur-échantillonnage.



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 diminuer 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 : 0.9099804305283757
Matrice de confusion :
[[925  47]
 [ 45   5]]
Rapport de classification :
              precision    recall  f1-score   support

           0       0.95      0.95      0.95       972
           1       0.10      0.10      0.10        50

    accuracy                           0.91      1022
   macro avg       0.52      0.53      0.53      1022
weighted avg       0.91      0.91      0.91      1022
```



4 Phase de modélisation

4.1 Préambule

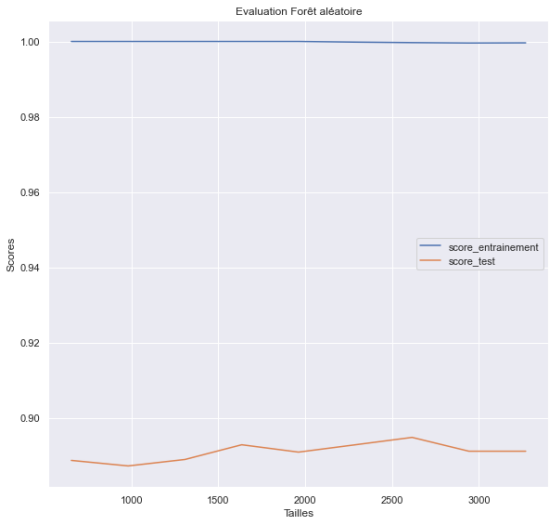
On devra entraîner un certain nombre de modèles et sélectionner celui qui donnera le meilleur score et qui ne sera ni trop complexe (score élevé sur les données d'entraînement et faible sur les données de test) et ni trop simple (score faible sur les données d'entraînement et de test). Le

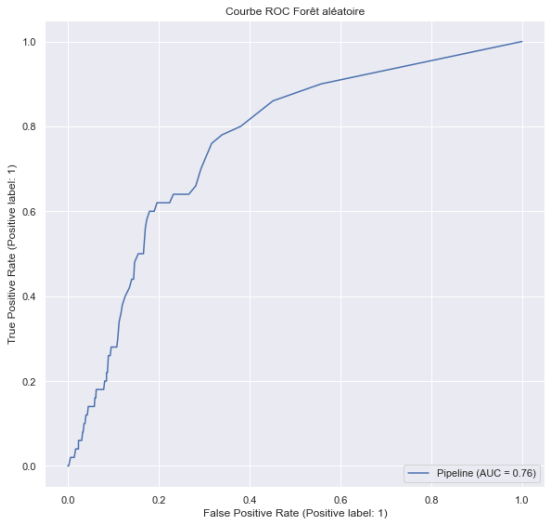
modèle final sera choisi parmi les suivantes : RandomForestClassifier, AdaBoostClassifier, SVC, KNeighborsClassifier et LogisticRegression.

4.2 Sélection de modèle

4.2.1 Random Forest

```
---------------
Pour le modèle Forêt aléatoire :
La precision du modèle : 0.8884540117416829
Matrice de confusion :
[[899  73]
 [ 41   9]]
Rapport de classification :
              precision    recall  f1-score   support

           0       0.96      0.92      0.94       972
           1       0.11      0.18      0.14        50

    accuracy                           0.89      1022
   macro avg       0.53      0.55      0.54      1022
weighted avg       0.91      0.89      0.90      1022
```

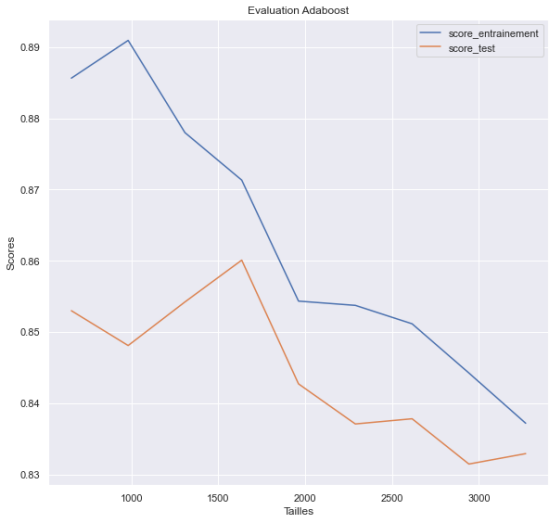


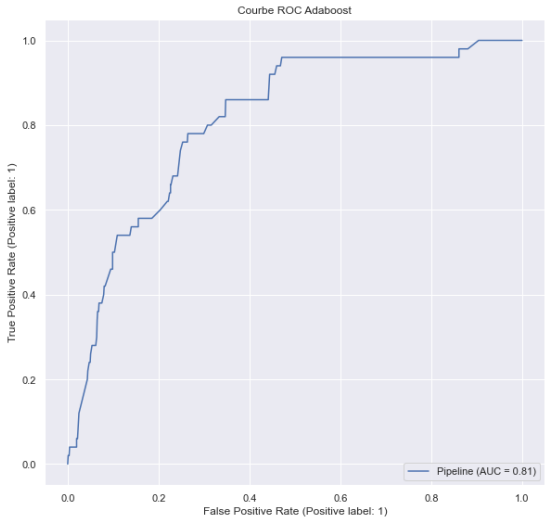


4.2.2 Adaboost

```
---------------
Pour le modèle Adaboost :
La precision du modèle : 0.8189823874755382
Matrice de confusion :
[[808 164]
```

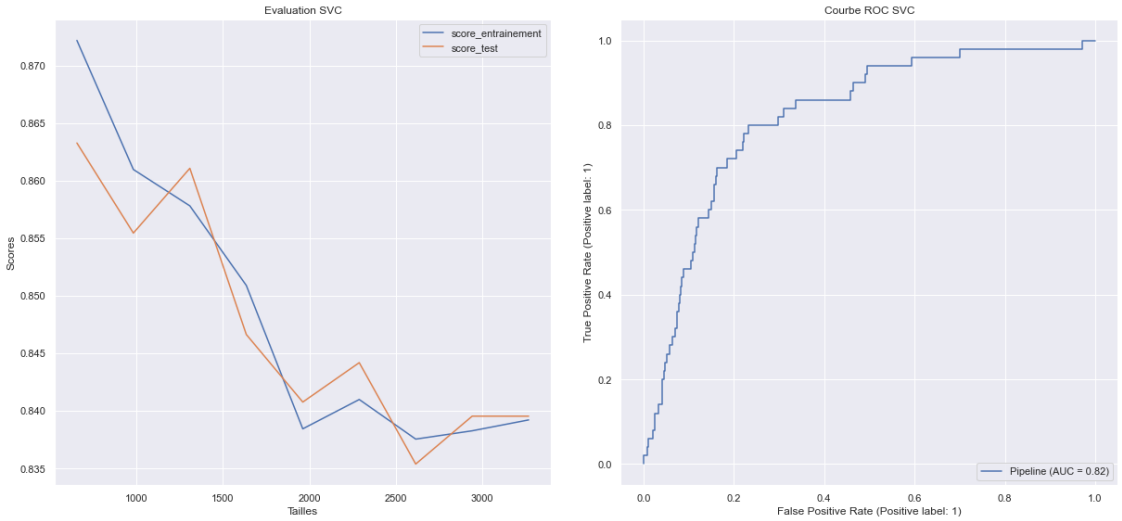
```
 [ 21  29]]
Rapport de classification :
              precision    recall  f1-score   support

           0       0.97      0.83      0.90       972
           1       0.15      0.58      0.24        50

    accuracy                           0.82      1022
   macro avg       0.56      0.71      0.57      1022
weighted avg       0.93      0.82      0.87      1022
```





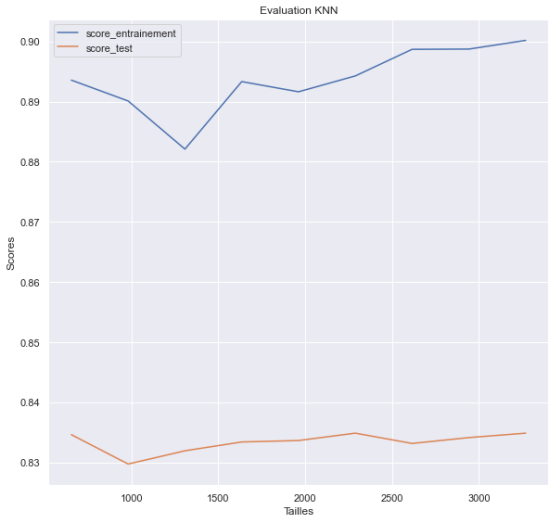
4.2.3 SVC

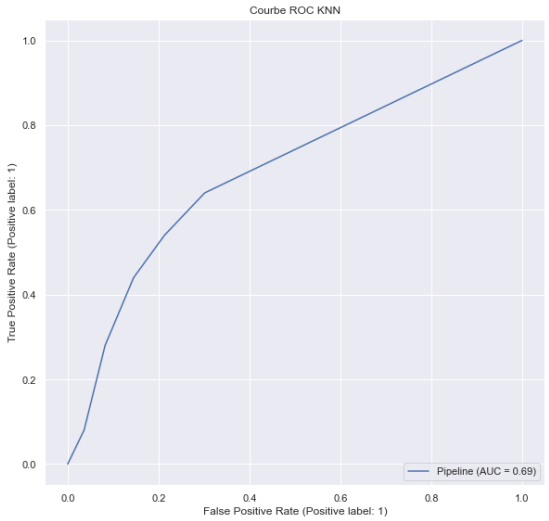
```
---------------
Pour le modèle SVC :
La precision du modèle : 0.8307240704500979
Matrice de confusion :
[[816 156]
 [ 17  33]]
Rapport de classification :
              precision    recall  f1-score   support

           0       0.98      0.84      0.90       972
           1       0.17      0.66      0.28        50

    accuracy                           0.83      1022
   macro avg       0.58      0.75      0.59      1022
weighted avg       0.94      0.83      0.87      1022
```



4.2.4 KNN

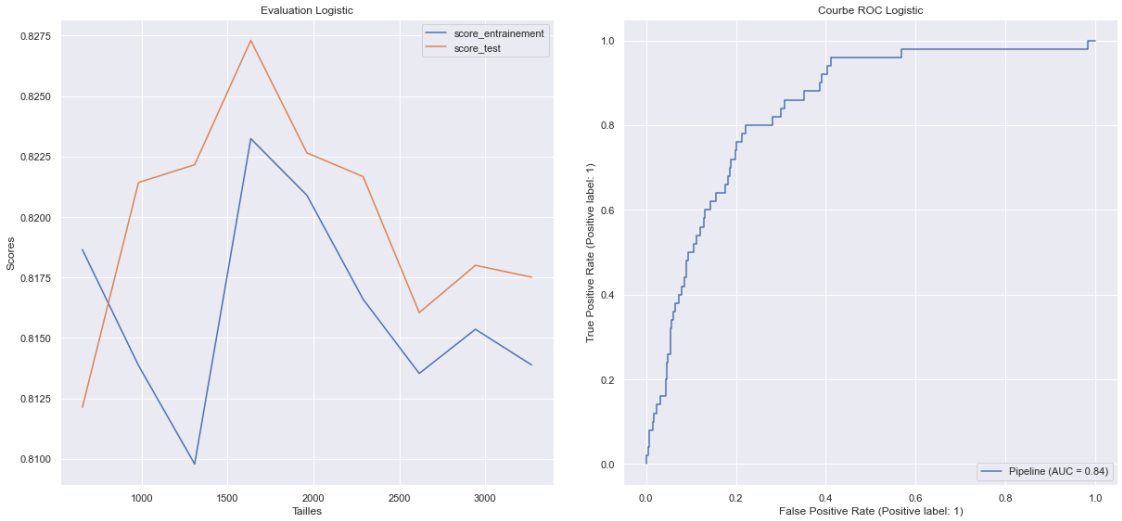
```
---------------
Pour le modèle KNN :
La precision du modèle : 0.8346379647749511
Matrice de confusion :
[[831 141]
 [ 28  22]]
Rapport de classification :
              precision    recall  f1-score   support

           0       0.97      0.85      0.91       972
           1       0.13      0.44      0.21        50

    accuracy                           0.83      1022
   macro avg       0.55      0.65      0.56      1022
weighted avg       0.93      0.83      0.87      1022
```





4.2.5 LOGISTIC

```
---------------
Pour le modèle Logistic :
La precision du modèle : 0.8101761252446184
Matrice de confusion :
[[794 178]
 [ 16  34]]
Rapport de classification :
              precision    recall  f1-score   support

           0       0.98      0.82      0.89       972
           1       0.16      0.68      0.26        50

    accuracy                           0.81      1022
   macro avg       0.57      0.75      0.58      1022
weighted avg       0.94      0.81      0.86      1022
```



Le meilleur modèle est donc la régression logistique avec une aire sous la courbe de 84%.

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.

[81]:
```
{'memory': None,
 'steps': [('columntransformer',
   ColumnTransformer(transformers=[('pipeline',
                                    Pipeline(steps=[('standardscaler',
                                                     StandardScaler()),
                                                    ('polynomialfeatures',
                                                     PolynomialFeatures())]),
<sklearn.compose._column_transformer.make_column_selector object at
0x000002A818B5CBB0>)])),
  ('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
0x000002A818B5CBB0>)]),
 '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
0x2a818b5cbb0>)],
 '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}
```

```
<ul>
    <li><strong>penalty (pénalité)</strong> : Ce paramètre est utilisé pour spécifier la norme
    <li><strong>dual (double)</strong> : Il est utilisé pour la formulation double ou primale.
    <li><strong>tol (tolérance)</strong> : Il représente la tolérance pour les critères d'arrêt
    <li><strong>C</strong> : Il représente l'inverse de la force de régularisation. Il doit tou
    <li><strong>fit_intercept</strong> : Ce paramètre spécifie si une constante (biais ou inter
    <li><strong>intercept_scaling</strong> : Ce paramètre est utile lorsque le solveur 'libline
```

```
    <li><strong>class_weight</strong> : Il représente les poids associés aux classes. Si nous u
    <li><strong>random_state</strong> : Ce paramètre représente la graine du nombre pseudo aléa
        <ul>
            <li><strong>int (entier)</strong> : dans ce cas, random_state est la graine utilis
            <li><strong>'Instance RandomState'</strong> : dans ce cas, random_state est le géné
            <li><strong>Aucun</strong> : dans ce cas, le générateur de nombres aléatoires est l
        </ul>
    </li>
    <li><strong>solver (solveur)</strong> : Ce paramètre représente l'algorithme à utiliser dan
        <ul>
            <li><strong>liblinear</strong> : C'est un bon choix pour les petits ensembles de do
            <li><strong>newton-cg</strong> : Il ne gère que la pénalité L2.</li>
            <li><strong>lbfgs</strong> : Pour les problèmes multiclasses, il gère la perte mult
            <li><strong>saga</strong> : C'est un bon choix pour les grands ensembles de données
            <li><strong>sag</strong> : Il est également utilisé pour les grands ensembles de do
        </ul>
    </li>
    <li><strong>max_iter</strong> : Comme son nom l'indique, il représente le nombre maximal d
    <li><strong>multi_class</strong> :
        <ul>
            <li><strong>ovr</strong> : Pour cette option, un problème binaire convient à chaque
            <li><strong>multimonial</strong> : Pour cette option, la perte minimisée est l'ajus
            <li><strong>auto</strong> : Cette option sélectionnera 'ovr' si solver = 'liblinear
        </ul>
    </li>
    <li><strong>verbose</strong> : Par défaut, la valeur de ce paramètre est 0, mais pour le so
    <li><strong>warm_start</strong> : Avec ce paramètre défini sur True, nous pouvons réutilise
    <li><strong>n_jobs</strong> : Si multi_class = 'ovr', ce paramètre représente le nombre de
    <li><strong>l1_ratio</strong> : Il est utilisé dans le cas où penalty = 'elasticnet'. Il s
</ul>
```

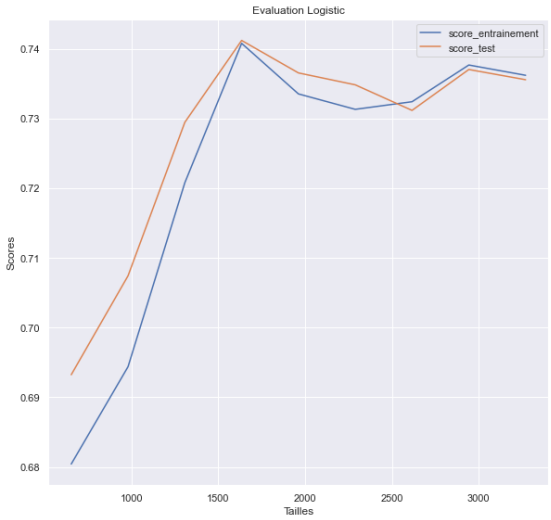
Nous utiliserons certaines de ces paramètres pour optimiser le modèle.

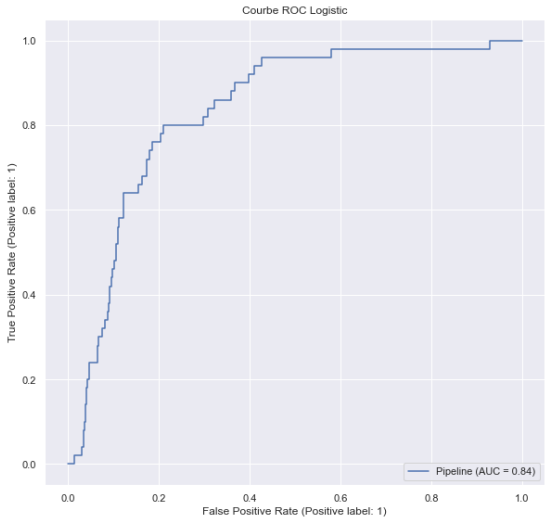
4.4 Optimisation de la performance du modèle

Optimisons la performance du modèle avec RandomizedSearchCV. On prendra comme métrique la sensibilité pour la recherche des meilleures paramètres du modèle.

[84]:
```
{'smote__sampling_strategy': 0.9,
 'smote__k_neighbors': 6,
 'logisticregression__solver': 'lbfgs',
 'logisticregression__penalty': 'none',
 'logisticregression__fit_intercept': False,
 'logisticregression__dual': False,
 'logisticregression__C': 3.9000000000000004,
 'columntransformer__pipeline__polynomialfeatures__degree': 3}

La precision du modèle : 0.7240704500978473
Matrice de confusion :
```

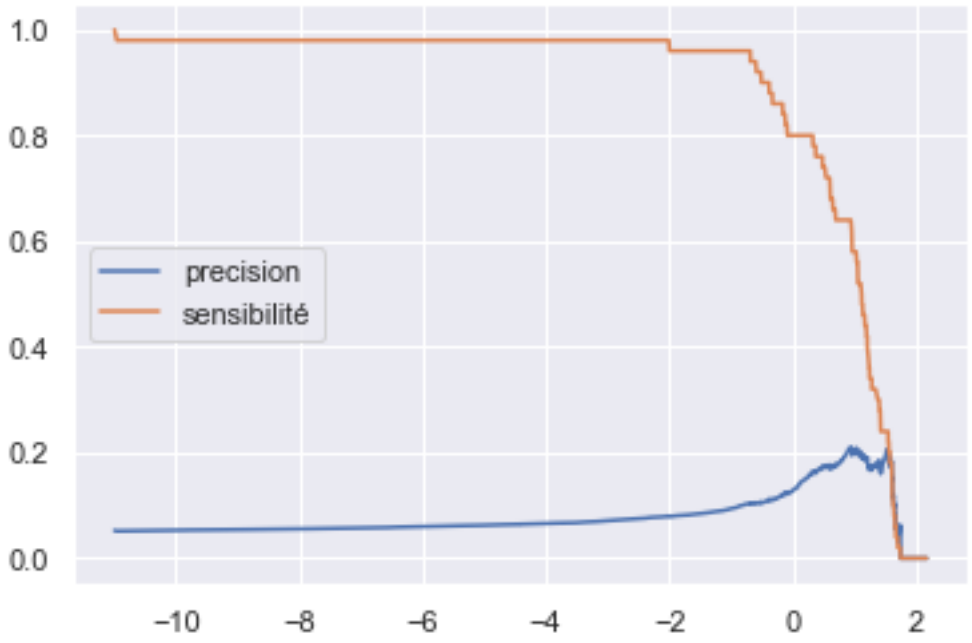
```
[[700 272]
 [ 10  40]]
Rapport de classification :
              precision    recall  f1-score   support

           0       0.99      0.72      0.83       972
           1       0.13      0.80      0.22        50

    accuracy                           0.72      1022
   macro avg       0.56      0.76      0.53      1022
weighted avg       0.94      0.72      0.80      1022
```





4.5 Compromis entre Sensibilité et Précision

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



Nous pouvons choisir un seuil de -0.5 pour obtenir une bonne sensibilité et diminuer la précision à l'inverse.

4.6 Conception du modèle final

Calcul de la sensibilité

[91]: 0.9

Nous obtenons ainsi une sensibilité finale de 90%