Institut de Formation et de Recherche en Informatique - UAC

Rapport de projet

Intitulé du projet : Développement de modèles d'apprentissage automatique supervisé

Enseignant:
Dr Ratheil HOUNDJI

Etudiant(e):
ADJIBADE A. Sadiyath M.

UE : Concepts de base de l'apprentissage automatique

Date de soumission: 02/07/20024

1. Contexte et Objectif

Ce projet est réalisé dans le cadre du cours «Concepts et applications de l'apprentissage automatique» et a pour objectif de construire des modèles d'apprentissage automatique en utilisant principalement la librairie Scikit-learn.

2. Partie 1

a. Description du dataset

Le dataset mis à notre disposition est un jeu de données de 1000 lignes et 7 colonnes.

Il présente 7 variables dont une dépendante (Les frais médicaux individuels facturés par l'assurance maladie (charges)) qui consistue la cible à prédire et est quantitative.

Parmi les 6 variables indépendantes, nous avons :
- 3 variables quantitatives (numériques) que sont : l'âge (age), l'indice de masse corporelle (bmi) et le nombre d'enfants couverts par l'assurance/nombre de personnes à charge (children); et
- 3 variables qualitatives (catégorielles) que sont : le sexe de l'assureur (sex), le statut de l'assuré, fumeur ou non (smoker) et la zone résidentielle du bénéficaire (region).

Les différents types sont répartis comme suit :
- int64 <- (age, children)
- float64 <- (bmi, charges)
- string <- (sex, smoker, region)

Parmi les variables catégorielles, 'sex' et 'smoker' sont binaires et 'region' est multiclasse.

b. Analyse descriptive

b.1 Analyse descriptive univariée des variables quantitatives

- 'age'

➢ Statistiques descriptives

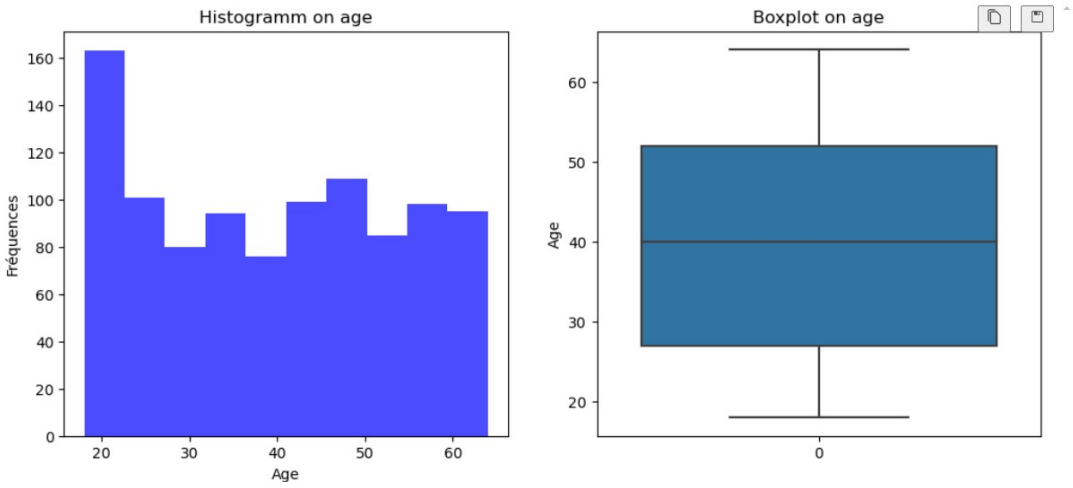
Tendance Centrale :
- Moyenne : 39,64. Ce qui signifie que l'âge moyen des assurés est d'environ 40 ans.

- Mode : 19. Ce qui signifie que l'âge le plus fréquent chez les assurés est de 19 ans.
- Médiane : 40. Ce qui signifie que la moitié des assurés a un âge inférieur à 40 ans, et l'autre moitié a un âge supérieur à 40 ans.

Position :
- Minimum : 18. Ce qui signifie que le plus jeune assuré a 18 ans.
- Maximum : 64. Ce qui signifie que l'assuré le plus âgé a 64 ans.
- Les valeurs de la colonne 'âge' sont comprises entre 18 et 64 ans.

Dispersion :
- Écart-type : 14,169. Ce qui signifie que la plupart des assurés ont un âge compris dans un intervalle de ±14 ans autour de la moyenne (soit entre 26 ans et 54 ans).
- Coefficient de Variation : 5,05. Ce qui indique que les âges varient considérablement par rapport à la moyenne.
- Premier Quartile (Q1) : 27. Ce qui signifie que 25% des assurés ont un âge inférieur ou égal à 27 ans.
- Deuxième Quartile (Q2) : 40. Correspond à la médiane.
- Troisième Quartile (Q3) : 52. Ce qui signifie que 75% des assurés ont un âge supérieur ou égal à 52 ans.
- Intervalle Inter-Quartile (IQR) : 25. Ce qui indique que les valeurs centrales des âges sont dispersées sur une plage de 25 ans, montrant une dispersion significative autour de la médiane.

➢ Visualisations



Interprétation

- Au niveau de l'histogramme, on remarque au début un pic atteignant 165, alors que le reste de la distribution est sensiblement uniforme. On déduit donc que la distribution est asymétrique.
Ce pic révèle que les assurés entre 18 et 22ans environ sont les plus fréquents tandis que ceux des autres tranches d'âges varient entre 70 et 110.

- Le boxplot, quant à lui, nous révèle l'absence d'outliers. On note également que les valeurs sont dispersés entre 18 et 64 ans.

• ‘bmi’

➢ Statistiques descriptives

D’après les recherches que nous avons menées:
- Si le bmi est inférieur à 18,5, on est probablement trop maigre.
- Si le bmi se situe entre 25 et 30, on a probablement quelques kilos en trop (surpoids).
- Si le bmi se situe entre 30 et 40, on est obèse.
- À partir d'un bmi de 40, on parle d'obésité morbide.

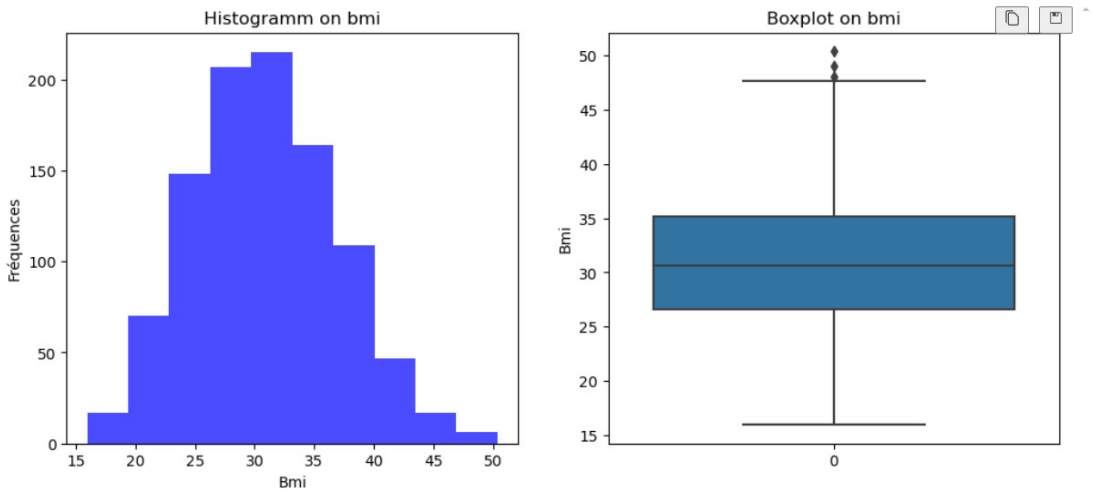
Ces informations nous permettent d’interprèterles différentes statistiques suivantes.

Tendance centrale:
- Moyenne : 30,86. Ce qui signifie que l'assuré moyen est en surpoids voire légèrement obèse.
- Mode : 32,3. Ce qui indique que le bmi le plus fréquent parmi les assurés correspond à une situation d'obésité.
- Médiane : 30,59. La moitié des assurés a un indice de masse corporelle inférieur à 30,59 et l'autre moitié a un indice supérieur à 30,59, montrant que plus de la moitié des assurés sont au moins en surpoids.

Position:
- Minimum : 15,96. Indiquant qu'il y a des assurés en sous-poids, le plus bas ayant un bmi de 15,96.
- Maximum : 50,38. Montrant également des cas d'obésité morbide parmi les assurés, le plus élevé ayant un bmi de 50,38.
- Les valeurs de la colonne 'bmi' sont comprises entre 15,96 et 50,38.

Dispersion:
- Écart-type : 6,05. La plupart des assurés ont un bmi qui se situe dans un intervalle de ±6 kg/m² autour de la moyenne (soit entre 24,81 et 36,91), montrant une prévalence significative de surpoids et d'obésité.
- Coefficient de Variation : 1,18. Indiquant que le bmi varie considérablement au sein de la population d'assurés de notre dataset.
- Premier Quartile (Q1) : 26,6. 25% des assurés ont un bmi inférieur ou égal à 26,6, montrant que moins d'un quart d'entre eux ne sont pas au minimum en surpoids.
- Deuxième Quartile (Q2) : 30,59. Correspond à la médiane.
- Troisième Quartile (Q3) : 35,11. 75% des assurés ont un bmi supérieur ou égal à 35,11.
- Intervalle Inter-Quartile (IQR) : 8,51. Les valeurs centrales du bmi s'étendent sur une plage d'environ 8 kg/m², montrant une concentration autour de la médiane.

➢ Visualisations



Interprétation

- Au niveau de l'histogramme, on remarque que les données ont globalement une forme de cloche, quoique légèrement asymétrique vers la droite. Elles suivent donc probablement une distribution assez proche d'une distribution normale.

- Le boxplot, quant à lui, nous révèle la présence de valeurs extrêmes entre 47 et 52. Toutefois, étant donnée que le bmi peut atteindre de telles valeurs, auquel cas on est dans de l'obésité, ces valeurs ne sont donc pas abérrantes. On note également que les valeurs sont très concentrées autour de la médiane.

• 'children'

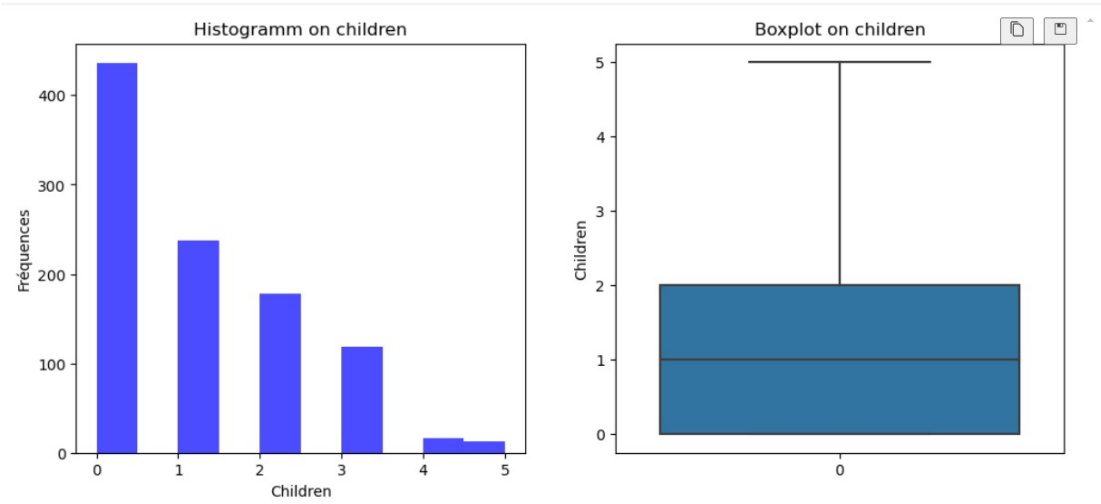
➢ Statistiques descriptives

Tendance centrale
- La moyenne est de 1,08, ce qui indique qu'en moyenne, les assurés ont un enfant ou une personne à charge couverts par l'assurance.
- Le mode est de 0, indiquant que la plupart des assurés n'ont pas d'enfants ou de personnes à charge couverts par l'assurance.
- La médiane est de 1, montrant que la moitié des assurés n'ont pas d'enfants ou de personnes à charge couverts par l'assurance, tandis que l'autre moitié a entre un et cinq enfants ou personnes à charge couverts.

Position
- Le minimum est de 0, indiquant qu'il y a des individus dont l'assurance couvre uniquement eux-mêmes.
- Le maximum est de 5, indiquant qu'il y a des individus dont l'assurance couvre jusqu'à cinq enfants ou personnes.

Dispersion
- L'écart-type est de 1,19, montrant que la plupart des individus ont entre 0 et 2 enfants ou personnes à charge couverts par l'assurance.
- Le coefficient de variation est de 1,33, indiquant que le nombre d'enfants/personnes à charge varie assez par rapport à la moyenne.
- Le premier quartile est de 0, montrant que 25% des individus n'ont pas d'enfants ou de personnes à charge couverts par l'assurance.
- Le deuxième quartile est de 1, correspondant à la médiane.
- Le troisième quartile est de 2, indiquant que 75% des assurés ont au moins deux enfants ou personnes à charge couverts par l'assurance.

➢ Visualisations



Interprétation

- Au niveau de l'histogramme, on remarque au début un pic atteignant 165, alors que le reste de la distribution est sensiblement uniforme. On déduit donc que la distribution est asymétrique.
Ce pic révèle que les assurés entre 18 et 22ans environ sont les plus fréquents tandis que ceux des autres tranches d'âges varient entre 70 et 110.

- Le boxplot, quant à lui, nous révèle l'absence d'outliers. On note également que les valeurs sont dispersés entre 18 et 64 ans.

• ‘bmi’

➢ Statistiques descriptives

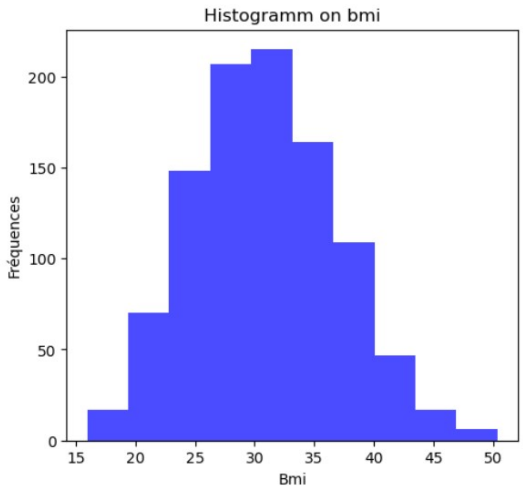
D’après les recherches que nous avons menées:
- Si le bmi est inférieur à 18,5, on est probablement trop maigre.
- Si le bmi se situe entre 25 et 30, on a probablement quelques kilos en trop (surpoids).
- Si le bmi se situe entre 30 et 40, on est obèse.
- À partir d'un bmi de 40, on parle d'obésité morbide.
Ces informations nous permettent d’interprèterles différentes statistiques suivantes.

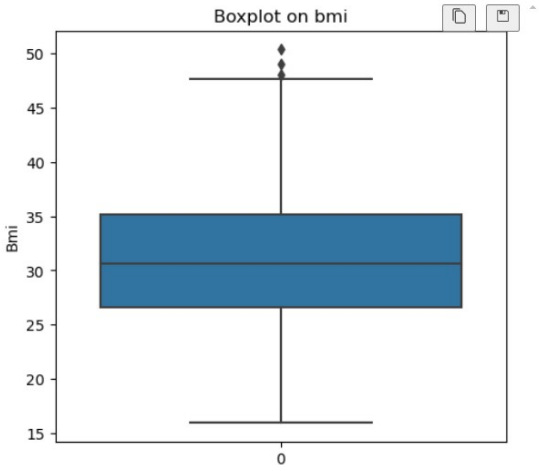
Tendance centrale:
- Moyenne : 30,86. Ce qui signifie que l'assuré moyen est en surpoids voire légèrement obèse.
- Mode : 32,3. Ce qui indique que le bmi le plus fréquent parmi les assurés correspond à une situation d'obésité.
- Médiane : 30,59. La moitié des assurés a un indice de masse corporelle inférieur à 30,59 et l'autre moitié a un indice supérieur à 30,59, montrant que plus de la moitié des assurés sont au moins en surpoids.

Position:
- Minimum : 15,96. Indiquant qu'il y a des assurés en sous-poids, le plus bas ayant un bmi de 15,96.
- Maximum : 50,38. Montrant également des cas d'obésité morbide parmi les assurés, le plus élevé ayant un bmi de 50,38.
- Les valeurs de la colonne 'bmi' sont comprises entre 15,96 et 50,38.

Dispersion:
- Écart-type : 6,05. La plupart des assurés ont un bmi qui se situe dans un intervalle de ±6 kg/m² autour de la moyenne (soit entre 24,81 et 36,91), montrant une prévalence significative de surpoids et d'obésité.
- Coefficient de Variation : 1,18. Indiquant que le bmi varie considérablement au sein de la population d'assurés de notre dataset.
- Premier Quartile (Q1) : 26,6. 25% des assurés ont un bmi inférieur ou égal à 26,6, montrant que moins d'un quart d'entre eux ne sont pas au minimum en surpoids.
- Deuxième Quartile (Q2) : 30,59. Correspond à la médiane.
- Troisième Quartile (Q3) : 35,11. 75% des assurés ont un bmi supérieur ou égal à 35,11.
- Intervalle Inter-Quartile (IQR) : 8,51. Les valeurs centrales du bmi s'étendent sur une plage d'environ 8 kg/m², montrant une concentration autour de la médiane.

➢ Visualisations





Interprétation

- Au niveau de l'histogramme, on remarque que les données ont globalement une forme de cloche, quoique légèrement asymétrique vers la droite. Elles suivent donc probablement une distribution assez proche d'une distribution normale.

- Le boxplot, quant à lui, nous révèle la présence de valeurs extrêmes entre 47 et 52. Toutefois, étant donnée que le bmi peut atteindre de telles valeurs, auquel cas on est dans de l'obésité, ces valeurs ne sont donc pas abérrantes. On note également que les valeurs sont très concentrées autour de la médiane.

• ‘charges’

➢ Statistiques descriptives

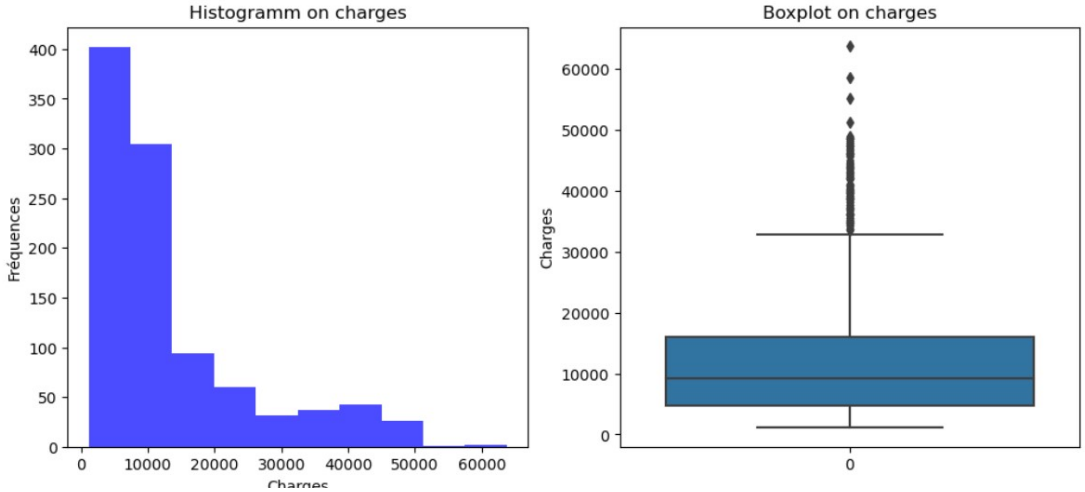
Tendance centrale
- La moyenne est de 13099,63, ce qui signifie que, en moyenne, les charges (primes d'assurance) des assurés s'élèvent à 13099,63.

- Le mode est de 1639,56, ce qui signifie que la charge la plus fréquente parmi les assurés est de 1639,56.
- La médiane est de 9286,85, ce qui signifie que la moitié des assurés a des charges inférieures à 9286,85 et l'autre moitié a des charges supérieures à 9286,85.

Position
- Le minimum est de 1121,87, cela signifie que le plus petit montant payé par un assuré pour ses charges est de 1121,87.
- Le maximum est de 63770,43, cela signifie que le plus grand montant payé par un assuré pour ses charges est de 63770,43.
- Les valeurs de la colonne 'charges' de ce dataset sont donc comprises entre 1121,87 et 63770,43. Cela suggère que ces valeurs varient probablement beaucoup et sont très étendues.

Dispersion
- L'écart-type est de 11994,13, ce qui signifie que la plupart des charges des assurés se situent dans un intervalle de ±11994,13 autour de la moyenne. Le dit intervalle étant [1105,5 et 25093,76].
- Le coefficient de variation est de 10981,93, ce qui montre que les charges varient beaucoup par rapport à la moyenne.
- Le premier quartile est de 4719,68, ce qui montre que 25% des assurés ont une charge inférieure ou égale à 4719,68.
- Le deuxième quartile est de 9286,85 et correspond à la médiane.
- Le troisième quartile est de 16073,10, ce qui signifie que 75% des assurés ont une charge inférieure ou égale à 16073,10.
- L'intervalle inter-quartile est de 11353,41, ce qui indique que les valeurs centrales des charges s'étendent sur une plage de 11353,41, montrant qu'elles sont assez dispersées autour de la médiane.

➢ Visualisations



Interprétation

- L'histogramme nous montre que la fréquence des charges diminue au fur et à mesure que le celles-ci augmentent, ce qui indique que moins de personnes paient des charges élevées.

- Le boxplot, quant à lui, nous révèle que la plupart des valeurs sont assez concentrées autour de la médiane. On note également la présence de valeurs extrêmes.

b.2 Analyse descriptive univariée des variables qualitatives

- ‘sex’

➢ Statistiques descriptives

Fréquences

 - Il y a 505 assurés de sexe masculin.

 - Il y a 495 assurés de sexe féminin.

Pourcentages

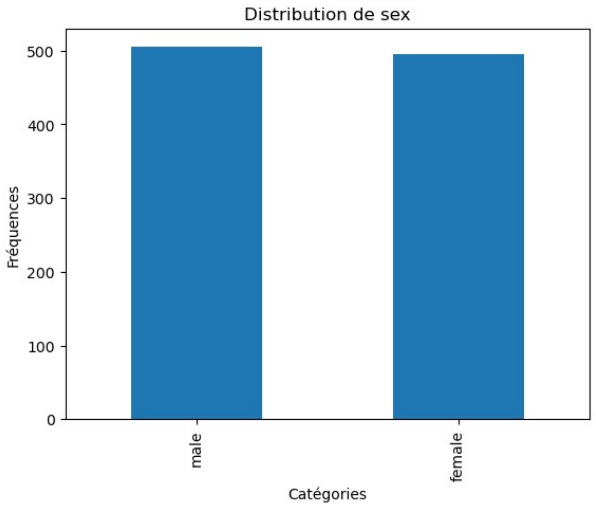
 - 50,5 % des assurés sont de sexe masculin.

 - 49,5 % des assurés sont de sexe féminin.

Interprètation

On note qu'il y a légèrement plus d'hommes (un peu plus de la moitié) que de femmes.

➢ Visualisations



- Ce graphique permet de confirmer qu'il y a pratiquement autant d'hommes que de femmes.

- ‘smoker’

➢ Statistiques descriptives

Fréquences

 - Il y a 803 assurés qui ne fument pas.

- - Il y a 197 assurés qui fument.

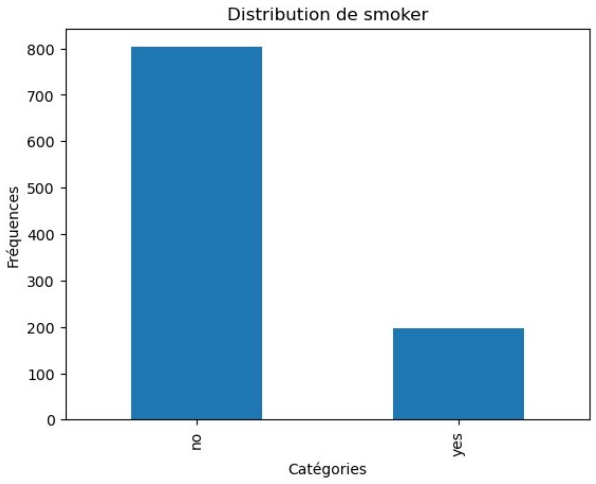
Pourcentages

 - 80,3 % des assurés ne fument pas.
 - 19,7 % des assurés fument.

Interprètation

On déduit que la plupart des assurés ne fument pas.

➢ Visualisations



- Ce graphique permet de confirmer qu'il y a une grande majorité de non fumeurs parmi les assurés.

• 'region'

➢ Statistiques descriptives

Fréquences :

- Il y a 278 assurés dans la région "southeast".
- Il y a 247 assurés dans la région "northeast".
- Il y a 244 assurés dans la région "southwest".
- Il y a 231 assurés dans la région "northwest".

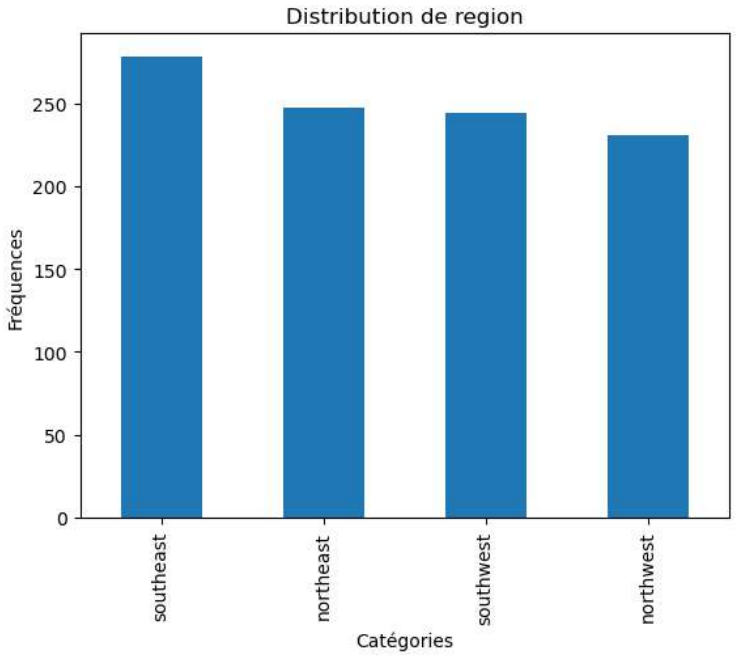
Pourcentages :

- 27,8 % des assurés sont dans la région "southeast".
- 24,7 % des assurés sont dans la région "northeast".
- 24,4 % des assurés sont dans la région "southwest".

Interprètation

Les régions "northeast", "southwest" et "northwest" ont des proportions assez similaires, variant de 23,1 % à 24,7 %. Seule "southeast" détonne avec une proportion un peu plus élevée (27,8%).

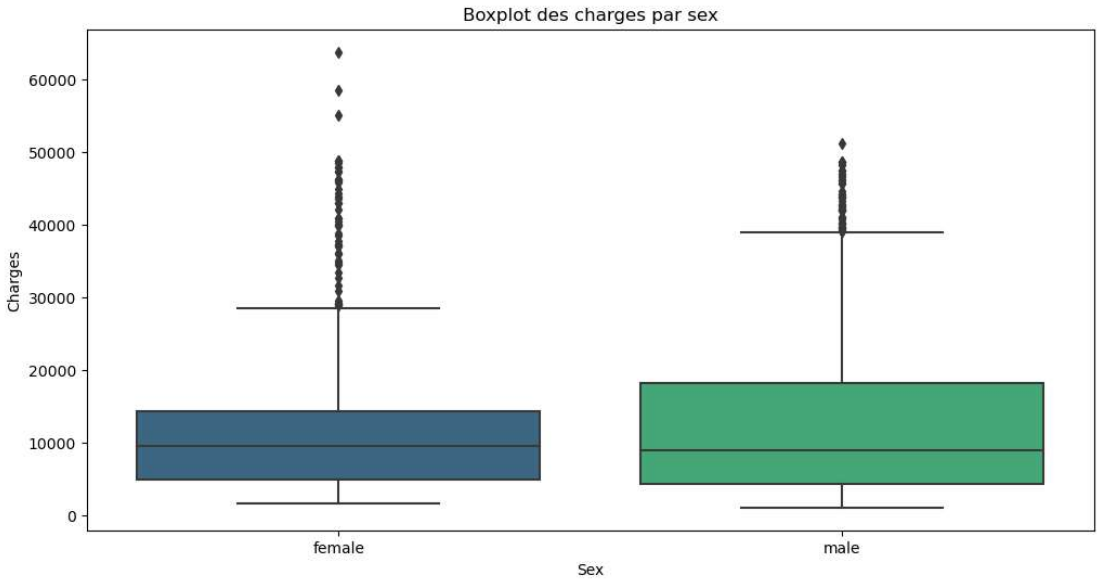
- Visualisations



- On remarque qu'il y a de manière générale, plus d'assurés à l'est qu'à l'ouest et au sud qu'au nord. Par rapport aux autres régions, on remarque qu'il y a plus d'assurés au sud-est(southeast).

b.2 Analyse descriptive bivariée entre la variable cible et les variables quantitatives

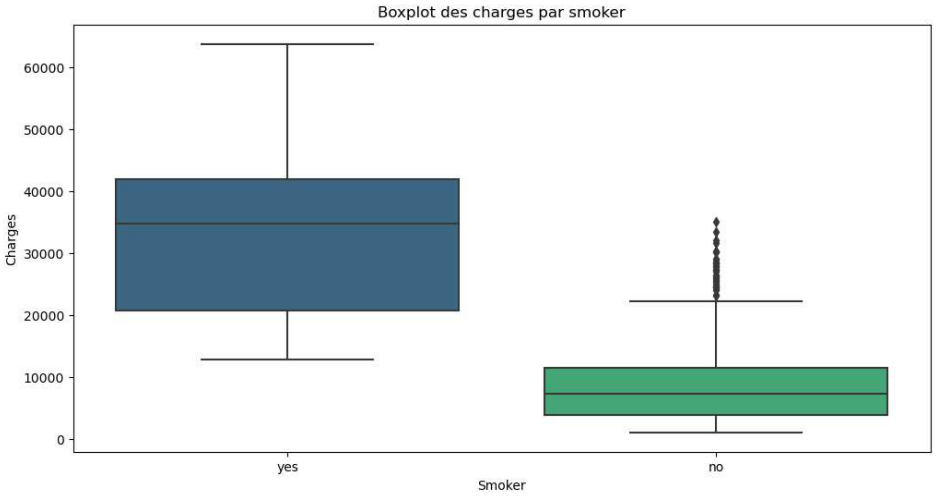
- Entre et ‘charges’ et ‘sex’



- Ce scatter montre qu'il n'y a pas vraiment de différences entre les charges payées par les hommes et celles payées par les femmes, à l'exception de quelques valeurs extremes.

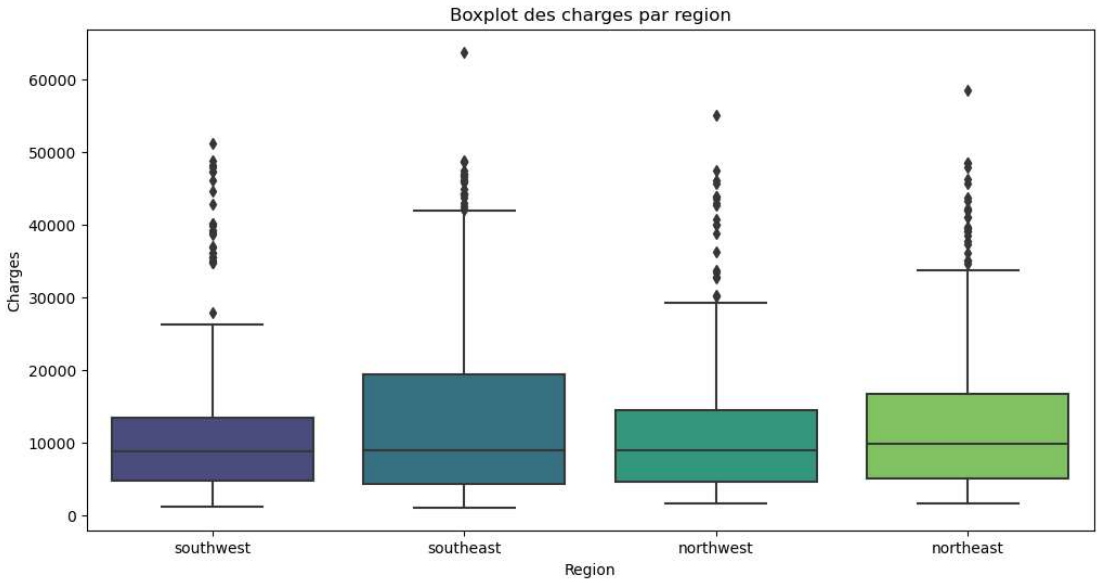
- Le sexe pourrait être donc une variable qui n'est pas très importante dans la prédiction des charges des assurés. Ou alors la relation qui lie la variable 'sex' à la variable 'charges' n'est pas linéaire.

• Entre et 'charges' et 'smoker'



- Ce scatter montre que les assurés fumeurs payent nettement pus de charges que ceux qui ne le sont pas. Cela révèle qu'ils sont probablement considérés comme à risque par les assurances. Un assuré fumeur payera donc plus cher pour son assurance maladie qu'un autre qui ne l'est pas.

• Entre et 'charges' et 'region'

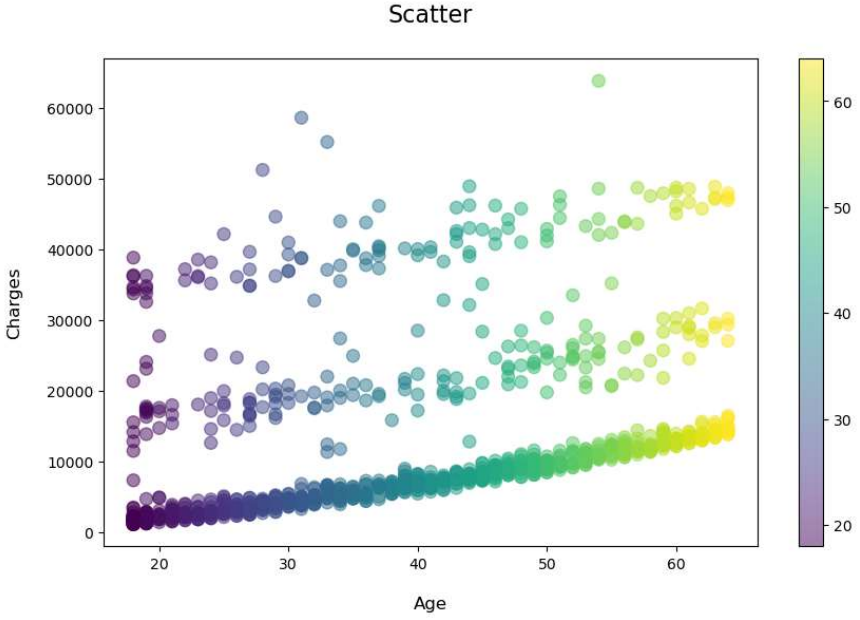


- Ce scatter montre que le montant de charges qu'on paye, selon qu'on vit au sud-ouest, au nord-ouest ou au nord-ouest, ne varie pas beaucoup. Par contre il varie significativement pour le sud-est, avec assez de valeurs extremes.

- Cela signifie que, étant donné que la répartition des charges est assez équitable entre ces trois régions, l'on pourrait s'en servir pour réduire le nombre de catégories de la variable region, ce qui permettrait d'optimiser les performances du modèle à modéliser sans perdre d'informations.

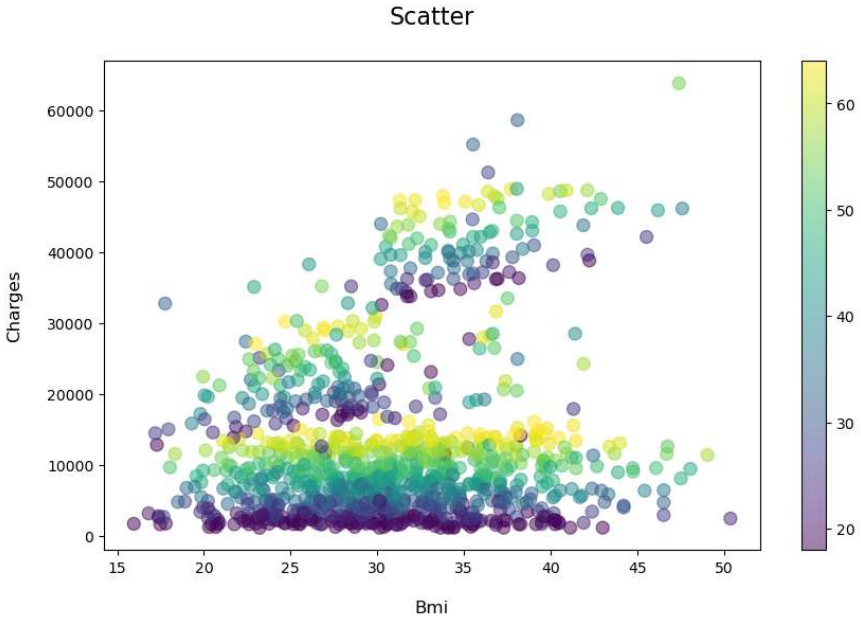
b.3 Analyse descriptive bivariée entre la variable cible et les variables quantitatives

- Entre et 'charges' et 'age'



- Ce scatter montre que, globalement, plus on est agé, plus on paye cher pour la prime d'assurance. Cela pourrait impliqué que plus l'assuré est agé, plus il est considéré comme risqué, à cause peut-etre de facteurs comme la santé.

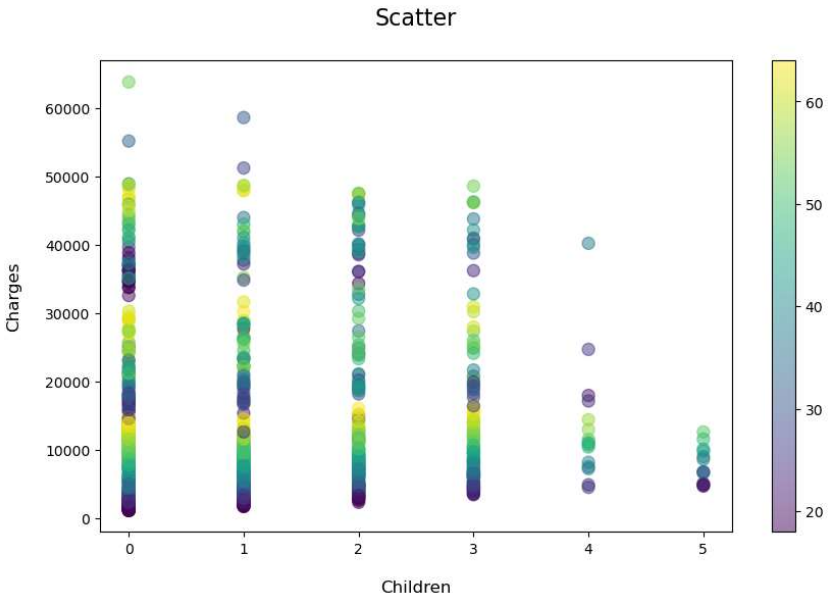
• Entre et 'charges' et 'bmi'



- Sur ce scatter, on peut remarque deux tendances. D'une part, les charges qui restent relativement faibles peu impote le bmi, et d'autre part, les charges qui augmentent au fur et à mesure que le bmi augmente.

- Cela pourrait impliquer que plus, l'assuré est en surpoids, plus il paye cher pour sa prime d'assurance.

• Entre et 'charges' et 'children'



- Ce scatter révèle que les charges ont tendance à baisser au fur et à mesure que le nombre augmente. Cela pourrait s'expliquer par la présence de formules familiales dans les différentes possibles de souscription à l'assurance maladie.

b.3 Matrice de corrélation

Interpretation

- Cette matrice révèle que nos données ne présentent pas de problème de multicolinéarité. Mais elle révèle également qu'il n'y a pas de corrélation forte entre les variables explicatives et la variable cible, la corrélation la plus forte n'étant que de 0.33 (entre l'age et les charges). Cela pourrait signifier qu'il existe entre nos variables des relations non linéaires.

b.4 Interpretation de l'ensemble des résultats

Cette analyse descriptive a permis de montrer que:

- Certaines variables comme 'smoker' ont une grande influence sur la target, qui varie beaucoup en fonction de celle-ci.

- Certaines autres variables comme 'sex' et dans une moindre mesure 'region' influencent moins la variance de la target. Les catégories de cette dernière peuvent d'ailleurs être regroupées de ce fait.

Globalement, on peut retenir que nos données présentent assez de valeurs extremes, que leur contexte ne permet pas de classer comme outliers, qu'elles ne présentant pas de problème de multicolinéarité également, et qu'on peut suspecter des relations non linéaires entre elles.

c. Pré-traitement des données

Afin de préparer les données pour la modélisation du modèles, diverses transfromations ont été appliquées.

- En premier lieu, nous avons recodé les modalités de la variable 'region'. En effet, étant donné qu'il a été remarqué dans l'analyse que les assurés sont pratiquement équitablement répartis, nous avons pensé que réduire le nombre de catégories de cette variable pourrait faire gagner le modèle en performances.

```python
# Recodons les modalités de la variable 'region'
def recode_region(region):
    if region in ['northwest', 'southwest']:
        return 'West'
    else:
        return 'East'

df['region'] = df['region'].apply(recode_region)
```

Python

Cette fonction appliquée au dataset permet d'effectuer cette transformation.

NB : La transformation Ouest-Est a été préférée à celle Nord - Sud car les deux ont été testées et les performances du modèle évaluées dans les deux cas, et il s'est avéré que celle-ci donnait les meilleurs performances.

- En second lieu, nous avons encodé les variables catégorielles afin qu'elles puissent être exploitées par l'algorithme qui nécessite des valeurs numériques.

```python
# Transformons les variables catégorielles en variables numériques
encoder = OneHotEncoder()

cat_var = ['sex', 'smoker', 'region']
for elt in cat_var:
    df[elt] = encoder.fit_transform(df[elt].values.reshape(-1, 1)).toarray()
```

Python

- Les données étant de différentes échelles (ex: age et children), nous les avons standardisé pour toutes les ramener à une moyenne de 0 et un écart-type de 1.

```python
# Mettons toutes les variables quantitatives à la meme échelle
scaler = StandardScaler()

var_a_standardiser = ['sex', 'smoker', 'region', 'age', 'bmi', 'children']
df[var_a_standardiser] = scaler.fit_transform(df[var_a_standardiser])
```

Python

Suite à cela, toutes les variables étant numériques, nous avons subdivisé nos données en ensembles entraînement et de test :

```python
data_train, data_test = train_test_split(df, test_size=0.2, random_state=42)
```

Puis nous avons séparé les features de la target :

```python
x_train  = data_train.drop(columns=['charges'])
y_train = data_train['charges']

x_test  = data_test.drop(columns=['charges'])
y_test = data_test['charges']
```

Les données sont maintenant pretes à être exploitées pour la modélisation.

d. Construction d'un premier modèle basé sur l'algorithme des k plus proches voisins

A cette étape, un premier modèle basé sur le KNeighborsRegressor a été créé, d'abord avec les hyper-paramètres par défaut, puis ensuite avec l'hyper-paramètre k configuré à 3, puis 7 et 10 pour finir. Les performances de ces modèles respectivement M0 M1 M2 M3 ont été évaluées et comparées.

	Modèles	Neighboors	MSE	R2 Score
0	M0 (par défaut)	5	2.875564e+07	0.852762
1	M1	3	2.859367e+07	0.853591
2	M2	7	2.926098e+07	0.850174
3	M3	10	2.740209e+07	0.859693

Conclusion : Le modèle M0 avec les paramètres par défaut, est le modèle plus performant.

e. Optimisation pour k=10

Dans cette partie, nous avons optimisé les hyperparamètres weight, p et algorithm du modèle M3 grace à la fonction suivante :

```python
def optimize_param(x_train, y_train):

    knn = KNeighborsRegressor(n_neighbors=10)

    param = {'weights': ['uniform', 'distance'],
    'p': [1, 2],
    'algorithm': ['ball_tree', 'kd_tree', 'brute', 'auto']}


    grid_search = GridSearchCV(knn, param, cv = 5, scoring = 'r2')

    grid_search.fit(x_train, y_train)

    best_model = grid_search.best_estimator_
    best_param = grid_search.best_params_
    best_score = grid_search.best_score_


    return best_model, best_param, best_score

best_knn, best_params, best_r2 = optimize_param(x_train, y_train)
best_knn.get_params(), best_r2
```

Il en est ressorti que les paramètres optimales pour ce modèle sont :

```
({'algorithm': 'brute',
  'leaf_size': 30,
  'metric': 'minkowski',
  'metric_params': None,
  'n_jobs': None,
  'n_neighbors': 10,
  'p': 2,
  'weights': 'distance'},
```

et son R2 score sur l'ensemble d'entraînement est de :

```
0.8156698208864304
```

L'évaluation de ses performances sur l'ensemble de test a donné les résultats suivants :

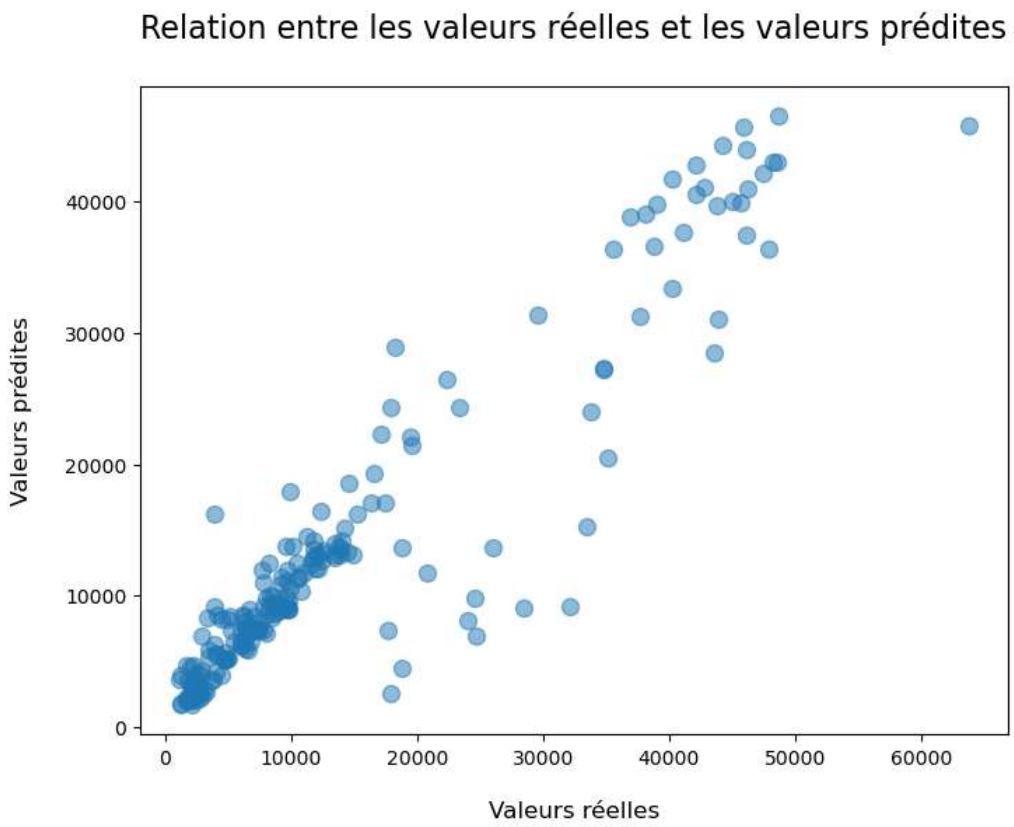
```
Performances du modèle optimisé

MSE:  26683728.438197374 \R2 score:  0.8633708210809945
```

Compte tenu du fait que le modèle a eu de meilleurs performances sur l'ensemble de test que l'ensemble entraînement, bien que l'écart ne soit pas énorme, on peut écarter l'hypothèse d'overfitting.

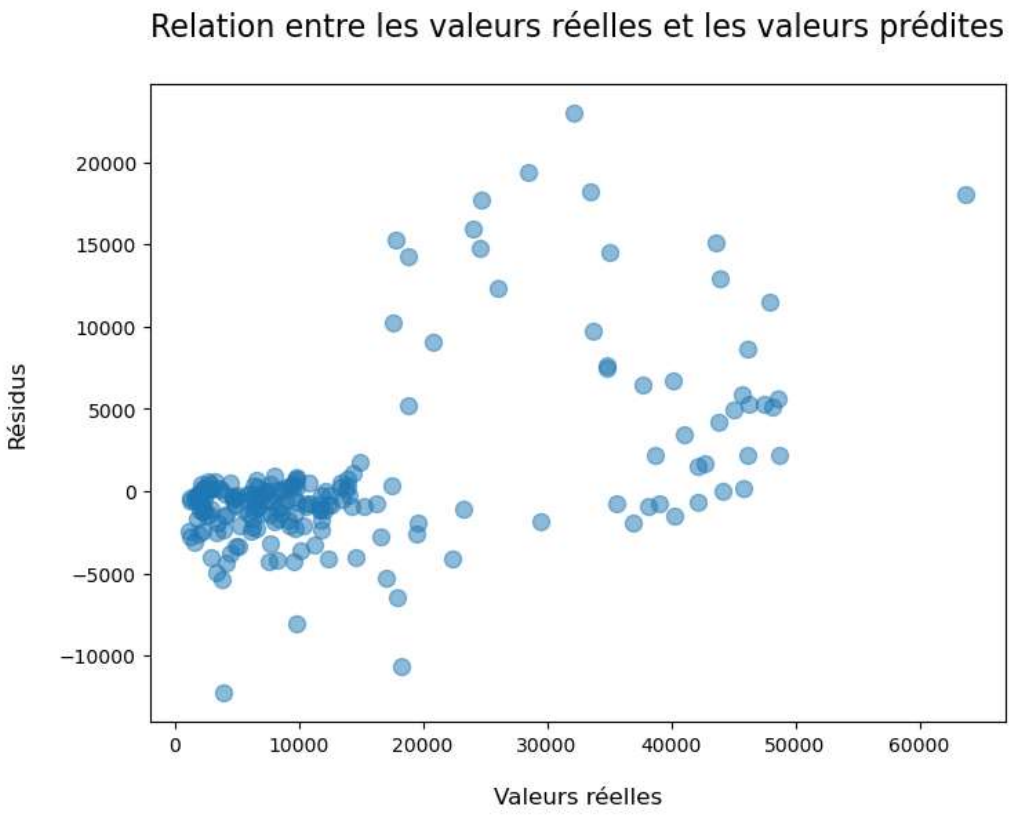
Les performances de ce modèle optimisé (MSE: 26683728.438197374, R2 score: 0.8633708210809945) étant clairement meilleures que celles du modèle par défaut M0 (MSE: 2.875564e+07, R2 score: 0.852762) , le modèle optimal est donc celui-ci, avec comme hyper-paramètres : {'algorithm': 'brute',
'leaf_size': 30,
'metric': 'minkowski',
'metric_params': None,
'n_jobs': None,
'n_neighbors': 10,
'p': 2,
'weights': 'distance'}

f. Analyse du meilleur modèle



Ce nuage de points montre que les valeurs prédites sont globalement assez proches des valeurs réelles étant donné que les points situent globalement le long de la ligne y=x, bien qu'ils soient un peu dispersés entre 18000 et 45000.

On peut donc conclure que le modèle est d'une performance acceptable.



On constate ici que, plus les valeurs de la variable cible augmente, plus les résidus augmentent également. Ces derniers sont assez dispersés.
Cela montre que le modèle a plus de mal à faire de bonnes prédictions, et donc est moins performant, sur les grandes valeurs que sur les petites, ce qui correspond à un problème d'hétéroscédasticité.

g. Synthèse générale de l'étude

- **Commentaires sur l'ensemble des résultats**

✧ L'analyse descriptive a révélé des tendances générales dans les données, ainsi que des valeurs extrêmes qui ne se révélées ne pas être des outliers.

✧ Le recodage des modalités de la variable 'region' a permis de réduire le nombre de catégorie de celle-ci.

✧ Le modèle par défaut s'est révélé plus performant. Après optimisation de tous les autres hyperparamètres également, seul l'hyperparamètre 'algorithme' a changé, mais la performance du modèle s'est encore plus améliorée, comme le montrent le R2 score (0.8633708210809945) et le MSE (2.875564e+07).

✧ L'analyse des relations entre valeurs réelles et prédites, ainsi qu'entre valeurs réelles et résidus, a mis en évidence une tendance à l'hétéroscédasticité, indiquant des erreurs de prédiction plus grandes pour les grandes valeurs de la variable cible.

- **Conclusion**

✧ Le modèle optimisé présente une performance satisfaisante dans la prédiction des valeurs de la variable cible, avec un R2 score élevé indiquant une bonne capacité à expliquer la variabilité des données.

✧ Toutefois, la présence d'hétéroscédasticité et de prédictions pas toujours précises ou exactes suggèrent qu'on pourrait encore l'améliorer.

• Quelques approches de solutions

+ Pour améliorer la précision du modèle, on pourrait collecter des données supplémentaires, éliminer les caractéristiques les moins pertinentes ou en créer de nouvelles qui apportent de nouvelles informations.

+ Pour réduire l'hétéroscédasticité pour améliorer les performances, on peut ajouter des variables explicatives qui pourraient aider le modèle à mieux comprendre la variabilité des données dans les catégories de données de la variable cible discriminées. On peut également effectuer des transformations comme la transformation logarthmique sur les données pour réduire voire résoudre ce problème.

h. Prédictions sur de nouvelles données

✧ La présence d'une colonne intruse 'a' sans données, sûrement due à une erreur de saisie a été détectée, et cette colonne a été supprimée.

✧ Les nouvelles données ont suivi le meme processus de pré-traitement que celles entraînement et de test (à l'exception du splitting et de la séparation d'après la target étant donné qu'elle n'est pas présente).

✧ Le modèle a été utilisé pour prédire sur ces données, et les prédictions obtenues ont été stockées dans un nouveau dataset nommé «Prédictions.xlsx».

3. Partie 2

Dans cette partie, nous avons testé d'autres modèles de régression suivant un protocole expérimental rigoureux.

• Le choix des modèles s'est basé sur le fait que, suite à l'entraînement du modèle précédent avec l'algorithme KNN Regressor, nous avons constaté la présence d'hétéroscédasticité, c'est-à-dire que plus les valeurs augmentent, plus le modèle a du mal à faire de bonnes prédictions. Cela laisse penser que la relation entre les variables n'est pas linéaire, et que le modèle a du mal à capturer cette complexité. De ce fait, nous avons décidé de choisir par la suite des modèles qui peuvent capturer des relations complexes et non linéaires :

- ✧ Le **Random Forest Regressor**, car c'est un modèle non linéaire qui peut capturer des relations complexes entre les variables.

- ✧ Le **Extreme Gradient Boosting Regressor (XGBoost)**, c'est un modèle qui a d'excellentes performances en terme de précision, et qui peut gérer des relations non linéaires complexes.

- ✧ Le **Support Vector Regression (SVR)**, car il peut capturer des relations non linéaires complexes à travers des noyaux.

Ces modèles présentent chacun un certain nombre de caractéristiques importantes qui justifient leur efficacité.

Random Forest Regressor

Le Random Forest Regressor est un modèle d'apprentissage automatique utilisé pour les tâches de régression. Voici ses caractéristiques importantes :

+ Ensemble de Décision Trees : Le modèle est composé de plusieurs arbres de décision (decision trees). Chaque arbre est construit à partir d'un échantillon aléatoire des données d'entraînement.

+ Bootstrap Aggregating (Bagging) : Les arbres sont construits en utilisant des échantillons bootstrap, ce qui signifie que chaque arbre est construit à partir d'un sous-ensemble aléatoire des données avec remplacement.

+ Combinaison des Prédictions : Les prédictions des différents arbres sont combinées (généralement par moyenne) pour produire la prédiction finale. Cela permet de réduire la variance du modèle et d'améliorer la précision.

+ Robustesse : En combinant plusieurs arbres, le modèle devient plus robuste aux outliers et au bruit dans les données par rapport à un arbre de décision unique.

+ Hyperparamètres : Le modèle permet le réglage des hyperparamètres qui incluent principalement :
  + le nombre d'arbres (n_estimators)
  + la profondeur des arbres (max_depth)
  + le nombre de variables par split (max_features)
  + la taille minimale des échantillons par feuille (min_samples_leaf).

+ Importance des Caractéristiques : Le modèle peut fournir une mesure de l'importance des différentes caractéristiques (features), ce qui est utile pour l'interprétabilité.

+ Parallélisation : Le processus de formation des arbres peut être parallélisé, ce qui permet de réduire le temps de calcul.

Extreme Gradient Boosting Regressor (XGBoost)

Le XGBoost (Extreme Gradient Boosting) Regressor est une implémentation optimisée et performante des algorithmes de boosting pour les tâches de régression. Voici ses caractéristiques importantes :

- Boosting en Gradient : XGBoost utilise le boosting en gradient pour améliorer la précision des modèles. Il construit des modèles faibles successifs, chaque nouveau modèle cherchant à corriger les erreurs des modèles précédents.

- Efficacité et Vitesse : Grâce à des optimisations avancées telles que la parallélisation des opérations, XGBoost est connu pour être extrêmement rapide et efficace, même sur des ensembles de données volumineux.

- Gestion des valeurs manquantes : XGBoost peut gérer les valeurs manquantes en les imputant de manière intelligente pendant la phase d'entraînement, ce qui permet de maintenir une bonne précision sans nécessiter un prétraitement intensif des données. Bien que cette caractéristique n'est pas vraiment importante ici vu que nos données ne contiennent pas de données mnquantes.

- Régularisation : Pour éviter le overfitting, XGBoost inclut des paramètres de régularisation L1 (lasso) et L2 (ridge), ce qui aide à généraliser le modèle et à améliorer sa robustesse.

- Importance des caractéristiques : XGBoost offre des outils intégrés pour évaluer l'importance des caractéristiques, ce qui permet d'identifier quelles variables ont le plus d'impact sur les prédictions du modèle.

- Support des arbres de décision : XGBoost construit des arbres de décision pour ses prédictions. Il propose différentes méthodes pour construire ces arbres, comme l'utilisation des pertes quadratiques ou logarithmiques.

- Traitement des données hétérogènes : XGBoost peut traiter différents types de données, y compris les données catégorielles, continues et ordinales.

- Compatibilité avec les GPU : XGBoost peut tirer parti des GPU pour accélérer les calculs, ce qui est particulièrement utile pour les ensembles de données très volumineux.

Support Vector Regression (SVR)

Le Support Vector Regressor (SVR) est une technique d'apprentissage automatique basée sur les principes des machines à vecteurs de support (SVM) utilisée pour la régression. Voici ses caractéristiques importantes :

-Principe de base : Le SVR vise à trouver une fonction qui dévie au maximum de toutes les données de formation par une marge ε et qui est aussi plate que possible.

- Fonction de coût : Le SVR utilise une fonction de coût qui ignore les erreurs qui sont dans une plage ε autour de la valeur prédite. Cela signifie qu'il y a une zone où les erreurs ne sont pas prises en compte.

- Kernel : Le SVR utiliser des fonctions kernel pour gérer des problèmes non linéaires en les transformant dans un espace de plus haute dimension où un hyperplan linéaire peut séparer les données. Les kernels courants sont le kernel linéaire, polynomial et RBF (Radial Basis Function).

- Hyperparamètres importants :
    - C (Regularization parameter) : Contrôle le compromis entre maximiser la marge et minimiser l'erreur de classification.
    - ε (Epsilon) : Détermine la largeur de la marge d'insensibilité dans laquelle les erreurs ne sont pas prises en compte.
    - Kernel parameters : Les paramètres spécifiques aux fonctions kernel utilisées, comme le degré du polynôme pour un kernel polynomial ou le paramètre gamma pour le kernel RBF.

- Complexité : Le SVR peut devenir complexe en termes de calcul et de mémoire pour des ensembles de données très larges, en particulier avec des kernels non linéaires.

- Robustesse : Le SVR est robuste aux outliers grâce à la marge ε qui ignore les erreurs mineures.

Par la suite, ces trois modèles ont été initialisés, optimisés et évalués .

Au niveau de l'optimisation, les modèles optimaux ont été :

- ✧ Le Random Forest Regressor dont les hyper-paramètres optimisés sont : {'max_depth': None, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 300}. Son R2 score sur l'ensemble d'entraînement a été de 0.8376588786025687.

- ✧ Le XGBoost dont les hyper-paramètres optimisés sont : {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 90, 'subsamples': 0.7}. Son R2 score sur l'ensemble d'entraînement a été de 0.8340573833572341.

- ✧ Le SVR dont les hyper-paramètres optimisés sont : {'C': 1000, 'epsilon': 0.01, 'kernel': 'poly'}. Son R2 score sur l'ensemble d'entraînement a été de 0.8132071857841938.

Après évaluation des performances de ces modèles optimaux sur l'ensemble de test, la hiérarchie des modèles par performances (du plus performant au moins performant) est la suivante:

1. Le RandomForestRegressor optimisé avec : MSE: 20479743.52111864, R2 score: 0.8951371976280205.

2. Le ExtremeGradientBoosting optimisé avec: MSE: 20539519.4617343, R2 score: 0.8948311257946053.

3. Le Support Vector Regression optimisé avec: MSE: 25012578.604578797 , R2 score: 0.8719276398983788.

4. Le KNeighborsRegressor optimisé avec: MSE: 26683728.438197374, R2 score: 0.8633708210809945.

Conclusion: Le meilleur modèle est donc le RandomForestRegressor .

4- Interface graphique

Nous avons implémenté une petite interface graphique qui prend de nouvelles entrées dans un formulaire et prédit la sortie en utilisant le meilleur modèle.

Technologies:

- Joblib pour le stockage et le chargement du modèle
- Streamlit pour l'interface graphique

Le modèle utilisé est le Random Forest Regressor optimisé.

Lien:

Fin du rapport.