

Département Génie Informatique

Cycle Ingénieur: LSI s3

Pr. *EL AACHAK LOTFI*

2024/2025



Atelier 1 : Régression linéaire

Élaborer par : *MOHAMED ILIASS KADDAR*

Module : Machine Learning

Introduction

L'objectif principal de cet atelier est de se familiariser avec les méthodes de régression en apprentissage automatique et d'appliquer ces techniques sur des jeux de données réels. Le but est d'apprendre à explorer des données, à préparer un jeu d'entraînement, à construire et comparer différents modèles de régression, puis à évaluer leurs performances à l'aide d'indicateurs quantitatifs. Ce travail permet de comprendre comment les modèles prédisent des valeurs continues et d'analyser leurs limites dans des contextes réels.

Dans le cadre de cet atelier, nous étudions trois familles de modèles de régression :

Régression linéaire simple : ce modèle cherche à établir une relation linéaire entre une seule variable explicative (feature) et une variable cible (target). Mathématiquement, il s'agit d'ajuster une droite qui minimise l'erreur entre les valeurs réelles et les valeurs prédites. Ce modèle est simple, facile à interpréter et constitue souvent une première approche pour vérifier si une relation linéaire existe entre deux variables.

Régression linéaire multiple : extension de la régression linéaire simple, ce modèle prend en compte plusieurs variables explicatives simultanément. La régression multiple permet de modéliser des phénomènes dont la cible dépend de plusieurs facteurs et nécessite souvent un prétraitement (encodage des variables catégorielles, standardisation des variables numériques, sélection de features).

Régression polynomiale : lorsque la relation entre les variables n'est pas linéaire, on peut transformer les features en fonctions polynomiales (puissances, interactions) et appliquer une régression linéaire sur ces nouvelles variables. La régression polynomiale permet de modéliser des courbes (non-linéaires) tout en conservant la formulation de régression linéaire sur les features transformées. Attention toutefois au risque de surapprentissage (overfitting) si le degré polynomial est trop élevé.

Les jeux de données utilisés pour cet atelier sont les suivants (sources) :

- **Expérience et Salaire** — dataset servant d'exemple classique pour la régression linéaire simple (Years of Experience vs Salary).

Source : [Kaggle](#).

- **Assurance** — dataset contenant des caractéristiques démographiques et médicales (age, sex, bmi, children, smoker, region) et la variable cible charges (coûts d'assurance). Utilisé pour la régression linéaire multiple.

Source : [Kaggle](#).

- **China GDP** — série temporelle du produit intérieur brut (GDP) de la Chine par année, utilisée pour comparer une régression linéaire simple et une régression polynomiale (modélisation de croissance non linéaire).

Source : [CognitiveClass / IBM](#)

Les outils employés pour réaliser cet atelier sont : **Python**, avec les bibliothèques **Pandas** pour la manipulation des données, **Matplotlib** et **Seaborn** pour la visualisation, et **Scikit-Learn** pour la mise en œuvre, l'entraînement et l'évaluation des modèles.

Partie 1 : Exploration & Visualisation des Données (EDA)

1.1. Chargement des données et Description statistique

On utilise Pandas pour charger les données depuis les datasets.

```
● ● ●
1 import pandas as pd
2
3 df1 = pd.read_csv(r'C:\Users\imk\Desktop\LSI\Maching learning\data sets\Salary_Data.csv')
4 df3 = pd.read_csv(r'C:\Users\imk\Desktop\LSI\Maching learning\data sets\insurance.csv')
```

Interprétation du dataset *df1* : Years & Salary

Le dataset **df1** contient 30 observations et deux variables numériques : les années d'expérience et le salaire correspondant. L'analyse exploratoire montre un jeu de données propre : aucune valeur manquante, aucune ligne dupliquée, et les deux variables sont parfaitement adaptées à une régression linéaire simple. Les statistiques descriptives indiquent une progression globale du salaire avec l'expérience, avec un salaire moyen d'environ **76 000 \$** et des valeurs comprises

entre 37 731 \$ et 122 391 \$, ce qui suggère une relation positive cohérente avec la théorie économique.

En résumé, df1 est un dataset propre, simple et idéal pour modéliser la relation linéaire entre expérience et salaire. La petite taille du dataset (30 lignes) limite la puissance prédictive d'un modèle.

Interprétation du dataset *df3 : Insurance dataset*

Le dataset df3 contient 1338 individus décrits par sept variables, mélangeant numériques (âge, BMI, enfants, charges) et catégorielles (sexe, fumeur, région). Les données sont complètes : aucune valeur manquante ni doublon. Sa taille importante le rend bien adapté à l'entraînement de modèles statistiques ou de machine learning. Les statistiques montrent une grande variabilité des charges d'assurance, influencées par des facteurs médicaux et personnels — notamment le statut de fumeur, l'âge et le BMI — qui sont des déterminants connus du coût de l'assurance santé.

En somme, df3 est un dataset propre, riche et suffisamment volumineux pour développer des modèles prédictifs fiables. Il permet d'analyser la manière dont les caractéristiques individuelles influencent les dépenses médicales, ce qui en fait un excellent jeu de données pour les tâches de régression multivariée.

```

1 df1 : years & salary dataset
2 Aperçu des premières lignes :
3
4    YearsExperience      Salary
5 0          1.1   39343.0
6 1          1.3   46205.0
7 2          1.5   37731.0
8 3          2.0   43525.0
9 4          2.2   39891.0
10 Informations générales :
11
12 RangeIndex: 30 entries, 0 to 29
13 Data columns (total 2 columns):
14 #   Column      Non-Null Count Dtype 
15 ---  -----      -----           ----- 
16 0   YearsExperience    30 non-null   float64
17 1   Salary        30 non-null   float64
18 dtypes: float64(2)
19 memory usage: 612.0 bytes
20 None
21
22 Nombre de lignes et colonnes : (30, 2)
23 Noms des colonnes : ['YearsExperience', 'Salary']
24 données statistiques
25
26    YearsExperience      Salary
27  count      30.000000  30.000000
28  mean       5.313333  76003.000000
29  std        2.837888  27414.429785
30  min        1.100000  37731.000000
31  25%       3.200000  56720.750000
32  50%       4.700000  65237.000000
33  75%       7.700000  100544.750000
34  max        10.500000 122391.000000
35 === Validation de la propriété du dataset ===
36
37 Valeurs manquantes par colonne :YearsExperience     0
38 Salary          0
39 dtype: int64

```

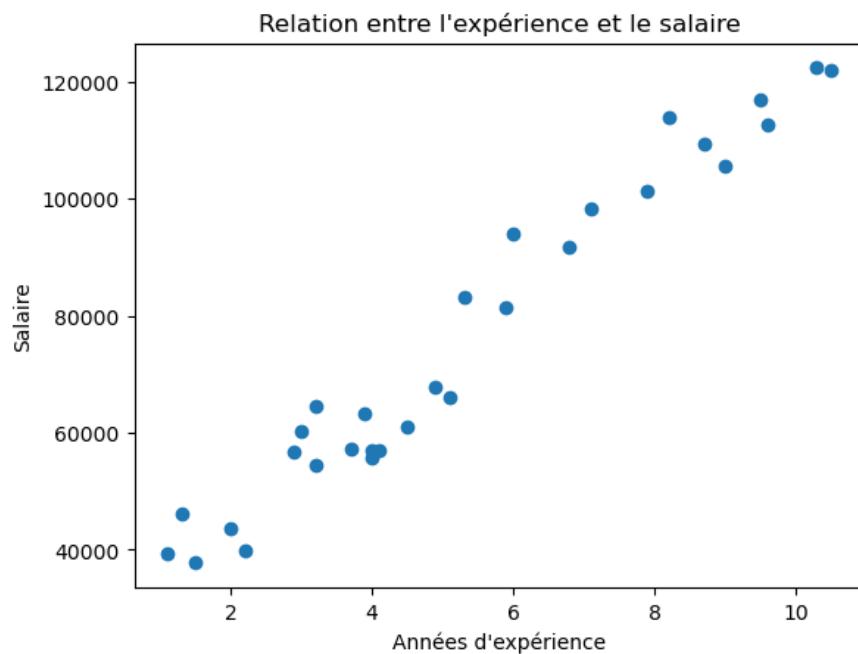
```

1 df3 : assurance dataset
2 Aperçu des premières lignes :
3
4    age      sex      bmi  children smoker  region  charges
5 0  19  female  27.900       0    yes  southwest  16884.92400
6 1  18    male  33.770       1     no  southeast  1725.55238
7 2  28    male  33.000       3     no  southeast  4449.46208
8 3  33    male  22.705       0     no  northwest  21984.47061
9
10 Informations générales :
11
12 RangeIndex: 1338 entries, 0 to 1337
13 Data columns (total 7 columns):
14 #   Column      Non-Null Count Dtype 
15 ---  -----      -----           ----- 
16 0   age         1338 non-null   int64 
17 1   sex         1338 non-null   object 
18 2   bmi         1338 non-null   float64
19 3   children    1338 non-null   int64 
20 4   smoker      1338 non-null   object 
21 5   region      1338 non-null   object 
22 6   charges     1338 non-null   float64
23 dtypes: float64(2), int64(2), object(3)
24 memory usage: 73.3+ KB
25 None
26
27 Nombre de lignes et colonnes : (1338, 7)
28 Noms des colonnes : ['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges']
29 données statistiques
30
31    age      bmi  children  charges
32  count  1338.000000  1338.000000  1338.000000
33  mean   39.207025  38.663397  1.094918  13278.422265
34  std    14.049960  6.098187  1.205493  12110.011237
35  min    18.000000  15.960000  0.000000  1121.873900
36  25%   27.000000  26.296250  0.000000  4740.287150
37  50%   39.000000  38.480000  1.000000  9382.033080
38  75%   51.000000  34.693750  2.000000  16639.912515
39  max    64.000000  53.130000  5.000000  63770.428010
40
41 Validation de la propriété du dataset
42
43    age      sex      bmi  children smoker  region  charges
44  count  1338.000000  1338.000000  1338.000000
45  mean   39.207025  38.663397  1.094918  13278.422265
46  std    14.049960  6.098187  1.205493  12110.011237
47  min    18.000000  15.960000  0.000000  1121.873900
48  25%   27.000000  26.296250  0.000000  4740.287150
49  50%   39.000000  38.480000  1.000000  9382.033080
50  75%   51.000000  34.693750  2.000000  16639.912515
51  max    64.000000  53.130000  5.000000  63770.428010
52
53 Validation de la propriété du dataset
54
55    age      sex      bmi  children smoker  region  charges
56  count  1338.000000  1338.000000  1338.000000
57  mean   39.207025  38.663397  1.094918  13278.422265
58  std    14.049960  6.098187  1.205493  12110.011237
59  min    18.000000  15.960000  0.000000  1121.873900
60  25%   27.000000  26.296250  0.000000  4740.287150
61  50%   39.000000  38.480000  1.000000  9382.033080
62  75%   51.000000  34.693750  2.000000  16639.912515
63  max    64.000000  53.130000  5.000000  63770.428010
64
65 Validation de la propriété du dataset
66
67    age      sex      bmi  children smoker  region  charges
68  count  1338.000000  1338.000000  1338.000000
69  mean   39.207025  38.663397  1.094918  13278.422265
70  std    14.049960  6.098187  1.205493  12110.011237
71  min    18.000000  15.960000  0.000000  1121.873900
72  25%   27.000000  26.296250  0.000000  4740.287150
73  50%   39.000000  38.480000  1.000000  9382.033080
74  75%   51.000000  34.693750  2.000000  16639.912515
75  max    64.000000  53.130000  5.000000  63770.428010
76
77 Validation de la propriété du dataset
78
79    age      sex      bmi  children smoker  region  charges
80  count  1338.000000  1338.000000  1338.000000
81  mean   39.207025  38.663397  1.094918  13278.422265
82  std    14.049960  6.098187  1.205493  12110.011237
83  min    18.000000  15.960000  0.000000  1121.873900
84  25%   27.000000  26.296250  0.000000  4740.287150
85  50%   39.000000  38.480000  1.000000  9382.033080
86  75%   51.000000  34.693750  2.000000  16639.912515
87  max    64.000000  53.130000  5.000000  63770.428010
88
89 Validation de la propriété du dataset
90
91    age      sex      bmi  children smoker  region  charges
92  count  1338.000000  1338.000000  1338.000000
93  mean   39.207025  38.663397  1.094918  13278.422265
94  std    14.049960  6.098187  1.205493  12110.011237
95  min    18.000000  15.960000  0.000000  1121.873900
96  25%   27.000000  26.296250  0.000000  4740.287150
97  50%   39.000000  38.480000  1.000000  9382.033080
98  75%   51.000000  34.693750  2.000000  16639.912515
99  max    64.000000  53.130000  5.000000  63770.428010
100
101 Validation de la propriété du dataset
102
103    age      sex      bmi  children smoker  region  charges
104  count  1338.000000  1338.000000  1338.000000
105  mean   39.207025  38.663397  1.094918  13278.422265
106  std    14.049960  6.098187  1.205493  12110.011237
107  min    18.000000  15.960000  0.000000  1121.873900
108  25%   27.000000  26.296250  0.000000  4740.287150
109  50%   39.000000  38.480000  1.000000  9382.033080
110  75%   51.000000  34.693750  2.000000  16639.912515
111  max    64.000000  53.130000  5.000000  63770.428010
112
113 Validation de la propriété du dataset
114
115    age      sex      bmi  children smoker  region  charges
116  count  1338.000000  1338.000000  1338.000000
117  mean   39.207025  38.663397  1.094918  13278.422265
118  std    14.049960  6.098187  1.205493  12110.011237
119  min    18.000000  15.960000  0.000000  1121.873900
120  25%   27.000000  26.296250  0.000000  4740.287150
121  50%   39.000000  38.480000  1.000000  9382.033080
122  75%   51.000000  34.693750  2.000000  16639.912515
123  max    64.000000  53.130000  5.000000  63770.428010
124
125 Validation de la propriété du dataset
126
127    age      sex      bmi  children smoker  region  charges
128  count  1338.000000  1338.000000  1338.000000
129  mean   39.207025  38.663397  1.094918  13278.422265
130  std    14.049960  6.098187  1.205493  12110.011237
131  min    18.000000  15.960000  0.000000  1121.873900
132  25%   27.000000  26.296250  0.000000  4740.287150
133  50%   39.000000  38.480000  1.000000  9382.033080
134  75%   51.000000  34.693750  2.000000  16639.912515
135  max    64.000000  53.130000  5.000000  63770.428010
136
137 Validation de la propriété du dataset
138
139    age      sex      bmi  children smoker  region  charges
140  count  1338.000000  1338.000000  1338.000000
141  mean   39.207025  38.663397  1.094918  13278.422265
142  std    14.049960  6.098187  1.205493  12110.011237
143  min    18.000000  15.960000  0.000000  1121.873900
144  25%   27.000000  26.296250  0.000000  4740.287150
145  50%   39.000000  38.480000  1.000000  9382.033080
146  75%   51.000000  34.693750  2.000000  16639.912515
147  max    64.000000  53.130000  5.000000  63770.428010
148
149 Validation de la propriété du dataset
150
151    age      sex      bmi  children smoker  region  charges
152  count  1338.000000  1338.000000  1338.000000
153  mean   39.207025  38.663397  1.094918  13278.422265
154  std    14.049960  6.098187  1.205493  12110.011237
155  min    18.000000  15.960000  0.000000  1121.873900
156  25%   27.000000  26.296250  0.000000  4740.287150
157  50%   39.000000  38.480000  1.000000  9382.033080
158  75%   51.000000  34.693750  2.000000  16639.912515
159  max    64.000000  53.130000  5.000000  63770.428010
160
161 Validation de la propriété du dataset
162
163    age      sex      bmi  children smoker  region  charges
164  count  1338.000000  1338.000000  1338.000000
165  mean   39.207025  38.663397  1.094918  13278.422265
166  std    14.049960  6.098187  1.205493  12110.011237
167  min    18.000000  15.960000  0.000000  1121.873900
168  25%   27.000000  26.296250  0.000000  4740.287150
169  50%   39.000000  38.480000  1.000000  9382.033080
170  75%   51.000000  34.693750  2.000000  16639.912515
171  max    64.000000  53.130000  5.000000  63770.428010
172
173 Validation de la propriété du dataset
174
175    age      sex      bmi  children smoker  region  charges
176  count  1338.000000  1338.000000  1338.000000
177  mean   39.207025  38.663397  1.094918  13278.422265
178  std    14.049960  6.098187  1.205493  12110.011237
179  min    18.000000  15.960000  0.000000  1121.873900
180  25%   27.000000  26.296250  0.000000  4740.287150
181  50%   39.000000  38.480000  1.000000  9382.033080
182  75%   51.000000  34.693750  2.000000  16639.912515
183  max    64.000000  53.130000  5.000000  63770.428010
184
185 Validation de la propriété du dataset
186
187    age      sex      bmi  children smoker  region  charges
188  count  1338.000000  1338.000000  1338.000000
189  mean   39.207025  38.663397  1.094918  13278.422265
190  std    14.049960  6.098187  1.205493  12110.011237
191  min    18.000000  15.960000  0.000000  1121.873900
192  25%   27.000000  26.296250  0.000000  4740.287150
193  50%   39.000000  38.480000  1.000000  9382.033080
194  75%   51.000000  34.693750  2.000000  16639.912515
195  max    64.000000  53.130000  5.000000  63770.428010
196
197 Validation de la propriété du dataset
198
199    age      sex      bmi  children smoker  region  charges
200  count  1338.000000  1338.000000  1338.000000
201  mean   39.207025  38.663397  1.094918  13278.422265
202  std    14.049960  6.098187  1.205493  12110.011237
203  min    18.000000  15.960000  0.000000  1121.873900
204  25%   27.000000  26.296250  0.000000  4740.287150
205  50%   39.000000  38.480000  1.000000  9382.033080
206  75%   51.000000  34.693750  2.000000  16639.912515
207  max    64.000000  53.130000  5.000000  63770.428010
208
209 Validation de la propriété du dataset
210
211    age      sex      bmi  children smoker  region  charges
212  count  1338.000000  1338.000000  1338.000000
213  mean   39.207025  38.663397  1.094918  13278.422265
214  std    14.049960  6.098187  1.205493  12110.011237
215  min    18.000000  15.960000  0.000000  1121.873900
216  25%   27.000000  26.296250  0.000000  4740.287150
217  50%   39.000000  38.480000  1.000000  9382.033080
218  75%   51.000000  34.693750  2.000000  16639.912515
219  max    64.000000  53.130000  5.000000  63770.428010
220
221 Validation de la propriété du dataset
222
223    age      sex      bmi  children smoker  region  charges
224  count  1338.000000  1338.000000  1338.000000
225  mean   39.207025  38.663397  1.094918  13278.422265
226  std    14.049960  6.098187  1.205493  12110.011237
227  min    18.000000  15.960000  0.000000  1121.873900
228  25%   27.000000  26.296250  0.000000  4740.287150
229  50%   39.000000  38.480000  1.000000  9382.033080
230  75%   51.000000  34.693750  2.000000  16639.912515
231  max    64.000000  53.130000  5.000000  63770.428010
232
233 Validation de la propriété du dataset
234
235    age      sex      bmi  children smoker  region  charges
236  count  1338.000000  1338.000000  1338.000000
237  mean   39.207025  38.663397  1.094918  13278.422265
238  std    14.049960  6.098187  1.205493  12110.011237
239  min    18.000000  15.960000  0.000000  1121.873900
240  25%   27.000000  26.296250  0.000000  4740.287150
241  50%   39.000000  38.480000  1.000000  9382.033080
242  75%   51.000000  34.693750  2.000000  16639.912515
243  max    64.000000  53.130000  5.000000  63770.428010
244
245 Validation de la propriété du dataset
246
247    age      sex      bmi  children smoker  region  charges
248  count  1338.000000  1338.000000  1338.000000
249  mean   39.207025  38.663397  1.094918  13278.422265
250  std    14.049960  6.098187  1.205493  12110.011237
251  min    18.000000  15.960000  0.000000  1121.873900
252  25%   27.000000  26.296250  0.000000  4740.287150
253  50%   39.000000  38.480000  1.000000  9382.033080
254  75%   51.000000  34.693750  2.000000  16639.912515
255  max    64.000000  53.130000  5.000000  63770.428010
256
257 Validation de la propriété du dataset
258
259    age      sex      bmi  children smoker  region  charges
260  count  1338.000000  1338.000000  1338.000000
261  mean   39.207025  38.663397  1.094918  13278.422265
262  std    14.049960  6.098187  1.205493  12110.011237
263  min    18.000000  15.960000  0.000000  1121.873900
264  25%   27.000000  26.296250  0.000000  4740.287150
265  50%   39.000000  38.480000  1.000000  9382.033080
266  75%   51.000000  34.693750  2.000000  16639.912515
267  max    64.000000  53.130000  5.000000  63770.428010
268
269 Validation de la propriété du dataset
270
271    age      sex      bmi  children smoker  region  charges
272  count  1338.000000  1338.000000  1338.000000
273  mean   39.207025  38.663397  1.094918  13278.422265
274  std    14.049960  6.098187  1.205493  12110.011237
275  min    18.000000  15.960000  0.000000  1121.873900
276  25%   27.000000  26.296250  0.000000  4740.287150
277  50%   39.000000  38.480000  1.000000  9382.033080
278  75%   51.000000  34.693750  2.000000  16639.912515
279  max    64.000000  53.130000  5.000000  63770.428010
280
281 Validation de la propriété du dataset
282
283    age      sex      bmi  children smoker  region  charges
284  count  1338.000000  1338.000000  1338.000000
285  mean   39.207025  38.663397  1.094918  13278.422265
286  std    14.049960  6.098187  1.205493  12110.011237
287  min    18.000000  15.960000  0.000000  1121.873900
288  25%   27.000000  26.296250  0.000000  4740.287150
289  50%   39.000000  38.480000  1.000000  9382.033080
290  75%   51.000000  34.693750  2.000000  16639.912515
291  max    64.000000  53.130000  5.000000  63770.428010
292
293 Validation de la propriété du dataset
294
295    age      sex      bmi  children smoker  region  charges
296  count  1338.000000  1338.000000  1338.000000
297  mean   39.207025  38.663397  1.094918  13278.422265
298  std    14.049960  6.098187  1.205493  12110.011237
299  min    18.000000  15.960000  0.000000  1121.873900
300  25%   27.000000  26.296250  0.000000  
```

1.2 Visualisation du nuage des points des deux data sets :

Dataset 1 : Expérience vs Salaire

Graphique scatter plot qui montre la relation entre l'expérience et le salaire



↗ Conclusion : relation linéaire claire

Dataset 2 : Assurance

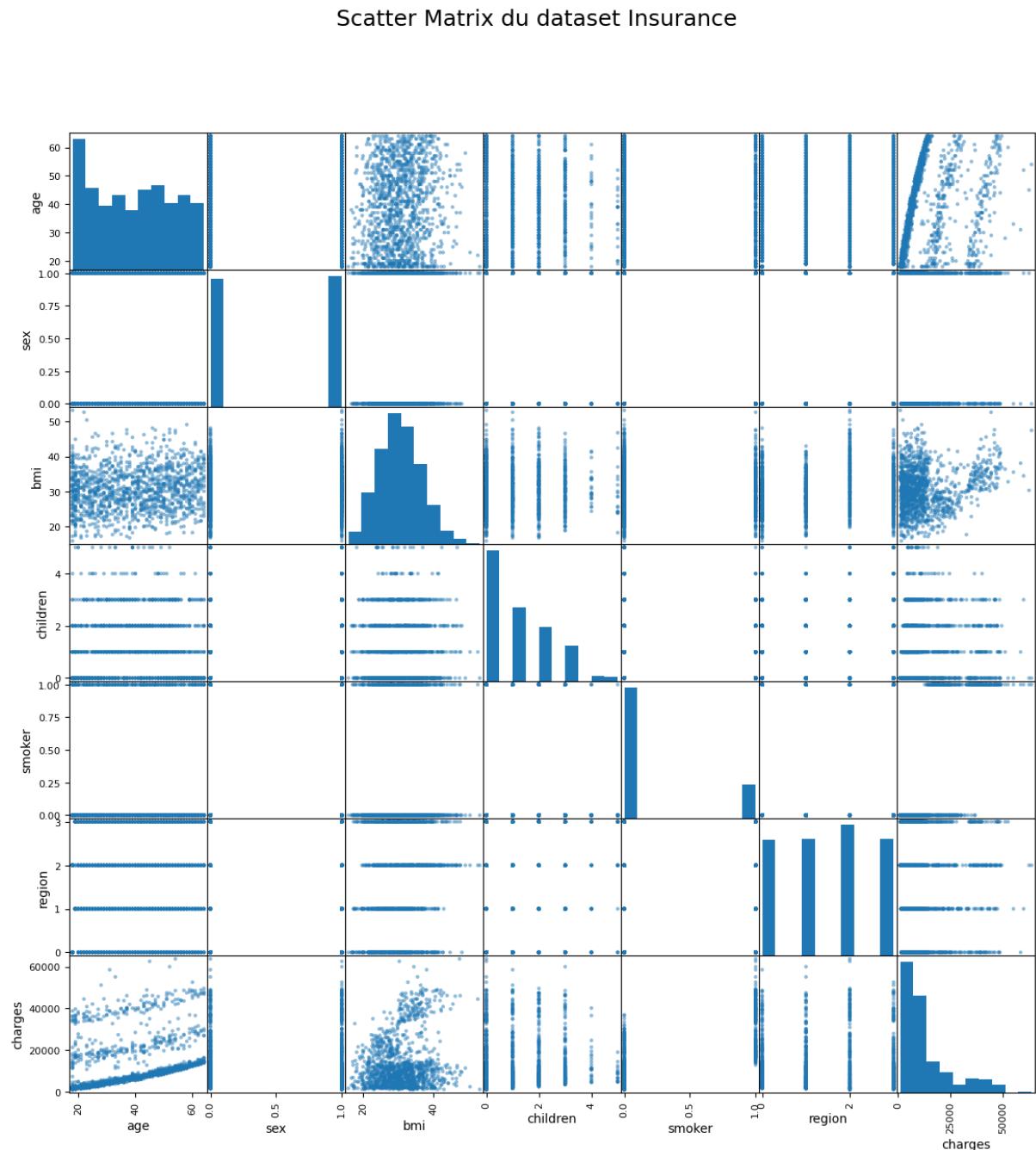
Encodage des variables catégorielles

```
1  from sklearn.preprocessing import LabelEncoder  
2  
3  encoder = LabelEncoder()  
4  df3['sex'] = encoder.fit_transform(df3['sex'])  
5  df3['smoker'] = encoder.fit_transform(df3['smoker'])  
6  df3['region'] = encoder.fit_transform(df3['region'])
```

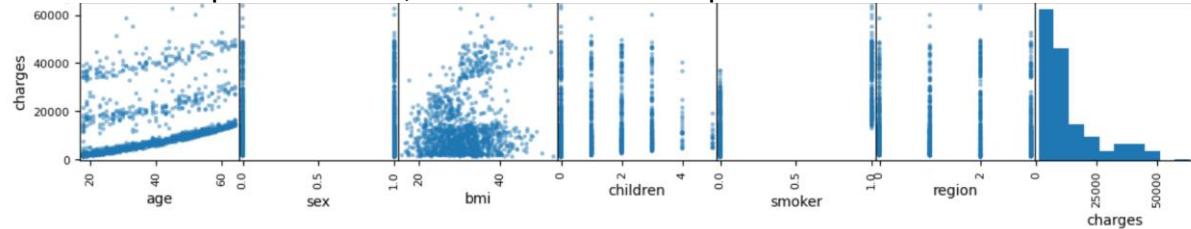
Certaines colonnes ne sont pas numériques, comme **sex**, **smoker**, ou **region**.
Le modèle linéaire ne peut pas traiter des chaînes de caractères : il faut donc les encoder.

- *smoker* : yes → 1, no → 0
- *sex* : male → 1, female → 0
- *region* : 1, 2, 3, 4

Scatter matrix du dataset Insurance, pour avoir un aperçu global du dataset.



La partie de la matrice que nous devons analyser est celle des variables en relation avec la charge. On peut prendre soit la partie horizontale, soit la partie verticale. Pour plus de clarté, nous choisissons la partie horizontale.



L'analyse montre que certaines variables ont une influence nettement plus marquée sur la charge. Le statut de fumeur (*smoker*) est le facteur le plus déterminant : les personnes qui fument (*smoker* = 1) présentent des charges beaucoup plus élevées que celles qui ne fument pas (*smoker* = 0). L'âge joue également un rôle important : plus l'âge augmente, plus la charge tend à croître. De même, l'indice de masse corporelle (*BMI*) montre une relation significative : lorsque le BMI augmente, les charges augmentent également, souvent de manière notable pour les valeurs élevées. En revanche, le sexe (homme = 1, femme = 0) ne montre pas de différence marquante sur la charge, tout comme le nombre d'enfants (*children*) et la région d'habitation (*region*), qui semblent exercer une influence faible ou négligeable sur les coûts. Ainsi, les facteurs les plus impactants restent principalement le tabagisme, l'âge et le BMI.

Partie 2 : Régression Linéaire Simple (Expérience /Salaire)

2.1 Séparation Train/Test

La séparation en ensembles *train* et *test* permet d'estimer la capacité d'un modèle à généraliser sur des données qu'il n'a pas vues pendant l'apprentissage. Si on entraînait et évaluait le modèle sur les mêmes données, on obtiendrait des performances optimistes (surapprentissage), et on ne saurait pas si le modèle marche sur de nouvelles observations.

***train* et *test* :**

- **Train** : sous-ensemble des données utilisé pour apprendre les paramètres du modèle.
- **Test** : sous-ensemble gardé de côté et utilisé uniquement pour évaluer la performance finale du modèle.

2.2 Entraînement du modèle :

On sépare les données en un ensemble d'entraînement et de test, puis crée un modèle de régression linéaire via **LinearReeression**. Le modèle est ensuite entraîné pour apprendre la relation entre l'expérience et le salaire.

on utilise souvent un ratio *80/20* ou *70/30* (train/test). On fixe random_state pour rendre la séparation reproductible, ici j'ai choisi 20% pour le test, via *test_size= 0.2*

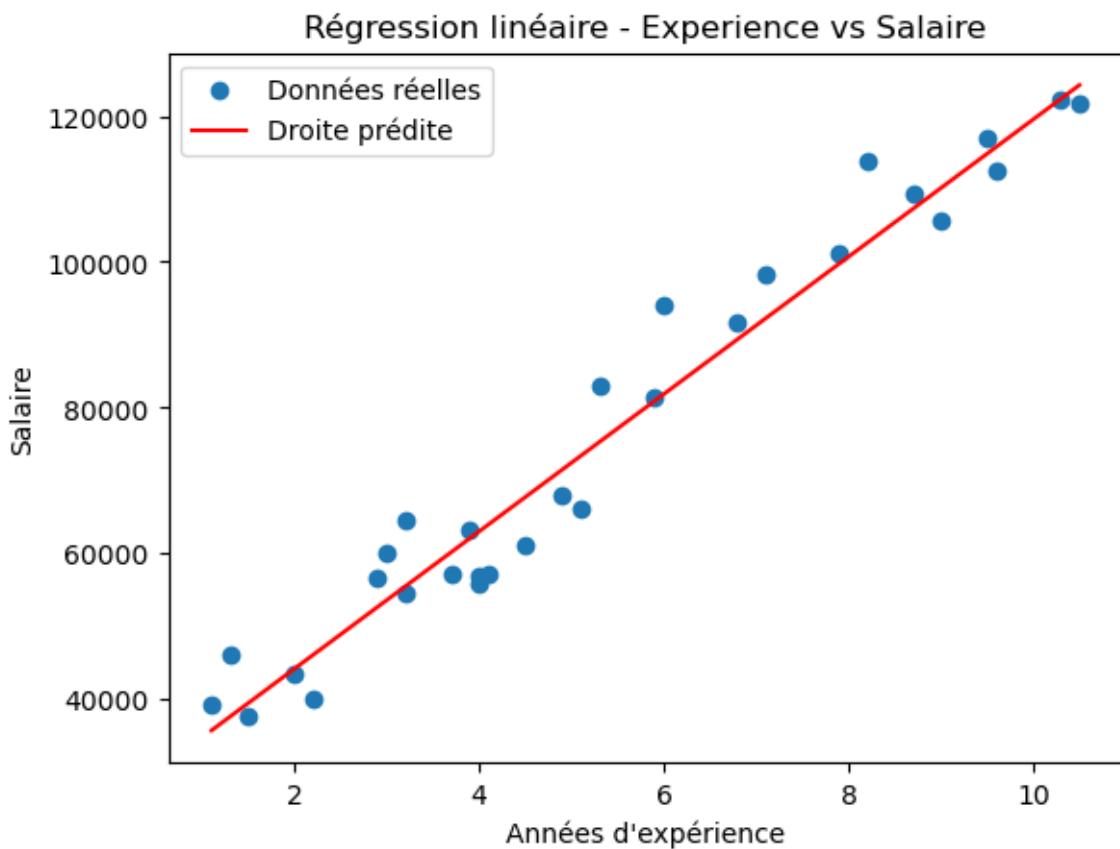
```
● ● ●  
1 from sklearn.linear_model import LinearRegression  
2 from sklearn.model_selection import train_test_split  
3  
4 X1 = df1['YearsExperience'].values.reshape(-1, 1)  
5 y1 = df1['Salary'].values  
6  
7 X1_train, X1_test, y1_train, y1_test = train_test_split(X1, y1, test_size=0.2, random_state=42)  
8  
9 modell= LinearRegression()  
10  
11 modell.fit(X1_train, y1_train)
```

2.3. Prédictions & Visualisation

On prédit sur l'ensemble de test pour mesurer la performance **réelle** du modèle : la comparaison *y_test* vs *y_pred* nous donne une estimation honnête de l'erreur sur des données invisibles lors de l'apprentissage. Si l'erreur sur le test est très supérieure à celle sur le train, c'est un signe de **surapprentissage**.

Droite de régression

Le traçage de la droite de régression sur le nuage de points permet de visualiser si le modèle capture bien la tendance générale.



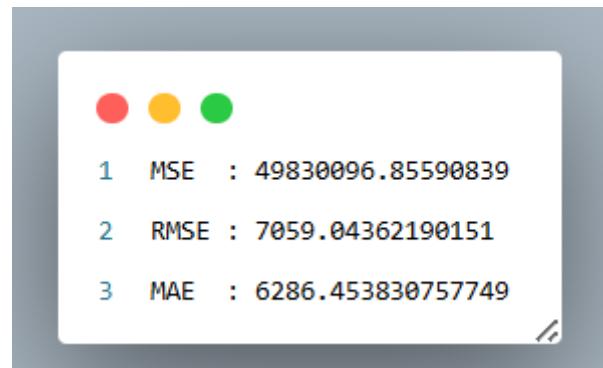
Interprétation visuelle

La droite suit bien la tendance des points, ce qui confirme la pertinence d'une relation linéaire. Avec la marge d'erreur est très faible.

2.4 Évaluation de la qualité du modèle :

Le modèle fonctionne presque correctement: Il prédit les salaires avec une erreur moyenne comprise entre 6200 et 7000 dh.

Mais il est clair que la marge d'erreur n'est pas plus faible.



❖ MAE = 6286.45

Le modèle commet en moyenne une erreur d'environ **6286 dh** sur ses prédictions. Cette valeur reste acceptable puisque les salaires du dataset se situent généralement entre **35 000 et 120 000 dh**. De plus, avec un dataset très réduit (**30 lignes seulement**), il est normal qu'un modèle n'arrive pas à apprendre des relations très précises.

❖ RMSE = 7059.04

La racine de l'erreur quadratique moyenne indique une erreur standard d'environ **7000 dh**.

Elle est naturellement plus élevée que le MAE car :

- le **RMSE pénalise davantage les grandes erreurs**,
- ce qui suggère la présence de quelques prédictions nettement éloignées de la valeur réelle.

❖ MSE = 49,830,096

Cette valeur semble élevée, mais c'est normal puisque :

- des erreurs de **6000 à 8000 dh** deviennent logiquement des millions lorsqu'on les élève au carré.
- C'est également pour cette raison que le **RMSE est plus interprétable** que le MSE.

2.5 Conclusion

L'erreur n'est pas plus faible parce que :

- le dataset est **très petit** (30 observations),
- la régression linéaire impose une **relation strictement linéaire**,
- le salaire ne dépend pas uniquement de l'expérience (niveau d'étude, entreprise, pays, etc.),
- le dataset contient des **outliers**.

L'erreur représente environ **10 %** du salaire moyen, ce qui est raisonnable pour un dataset aussi limité.

Partie 3 : Régression Linéaire Multiple (Assurance)

3.1. Nettoyage + EDA

Dans cette partie, nous analysons le dataset *Insurance* afin de comprendre la structure des données et préparer la régression linéaire multiple

The screenshot shows two code cells in a Jupyter Notebook environment. The left cell displays statistical summary and dimensionality information, while the right cell details column types, memory usage, and missing values.

Left Cell Output:

```
1 Résumé statistique :
2
3   count    age        sex      bmi   children   smoker \\
4   mean     39.222139  0.504862  30.663452  1.095737  0.204936
5   std      14.044333  0.500163  6.100468  1.205571  0.403806
6   min      18.000000  0.000000  15.960000  0.000000  0.000000
7   25%     27.000000  0.000000  26.290000  0.000000  0.000000
8   50%     39.000000  1.000000  30.400000  1.000000  0.000000
9   75%     51.000000  1.000000  34.700000  2.000000  0.000000
10  max      64.000000  1.000000  53.130000  5.000000  1.000000
11
12   region   charges
13   count    1337.000000 1337.000000
14   mean     1.516081 13279.121487
15   std      1.105208 12110.359656
16   min      0.000000 1121.873900
17   25%     1.000000 4746.344000
18   50%     2.000000 9386.161300
19   75%     2.000000 16657.717450
20   max      3.000000 63770.428010
21
22 dimensions :
23 (1337, 7)
```

Right Cell Output:

```
1 Types des colonnes :
2
3 Index: 1337 entries, 0 to 1337
4 Data columns (total 7 columns):
5 #   Column      Non-Null Count Dtype 
6 ---  --          -----          --  
7 0   age         1337 non-null   int64 
8 1   sex         1337 non-null   int64 
9 2   bmi         1337 non-null   float64
10 3  children    1337 non-null   int64 
11 4  smoker      1337 non-null   int64 
12 5  region      1337 non-null   int64 
13 6  charges     1337 non-null   float64
14 dtypes: float64(2), int64(5)
15 memory usage: 83.6 KB
16 None
17
18 valeurs manquantes :
19 age      0
20 sex      0
21 bmi      0
22 children 0
23 smoker   0
24 region   0
25 charges  0
26 dtype: int64
27
28 lignes dupliquées :
29 0
```

❖ Matrice de corrélation

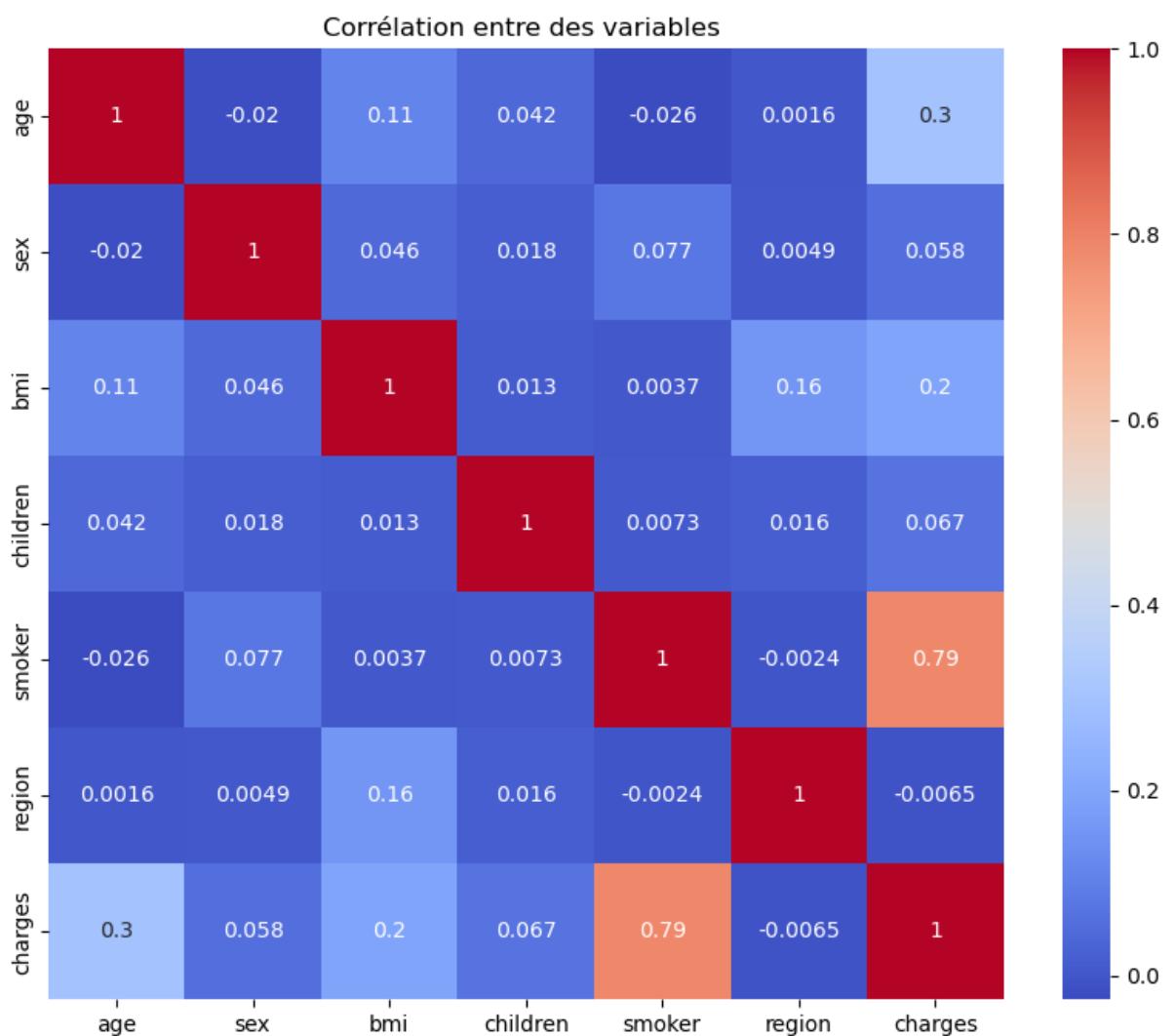
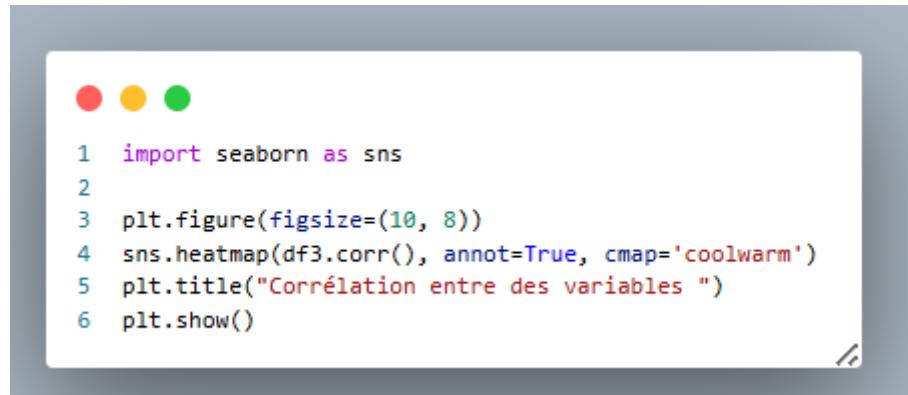
La **corrélation** permet d'identifier les variables qui ont la plus forte relation avec la target charges.

Dans notre dataset, on observe que :

- **smoker** est la variable la plus corrélée positivement avec les frais médicaux

- **bmi** montre une corrélation modérée
- **age** présente une corrélation positive importante

Ces trois caractéristiques ressortent fortement dans la matrice de corrélation.



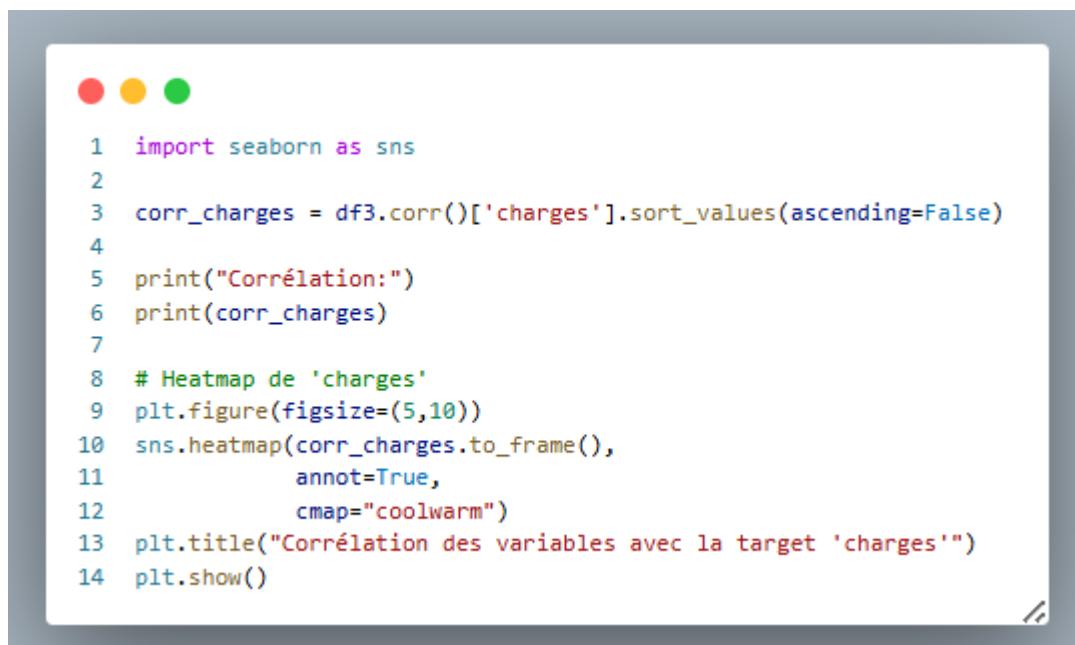
- Conclusion EDA

L'EDA montre clairement que certaines variables ont un impact dominant. Cela guidera la sélection des 3 features.

3.2. Sélection des 3 features :

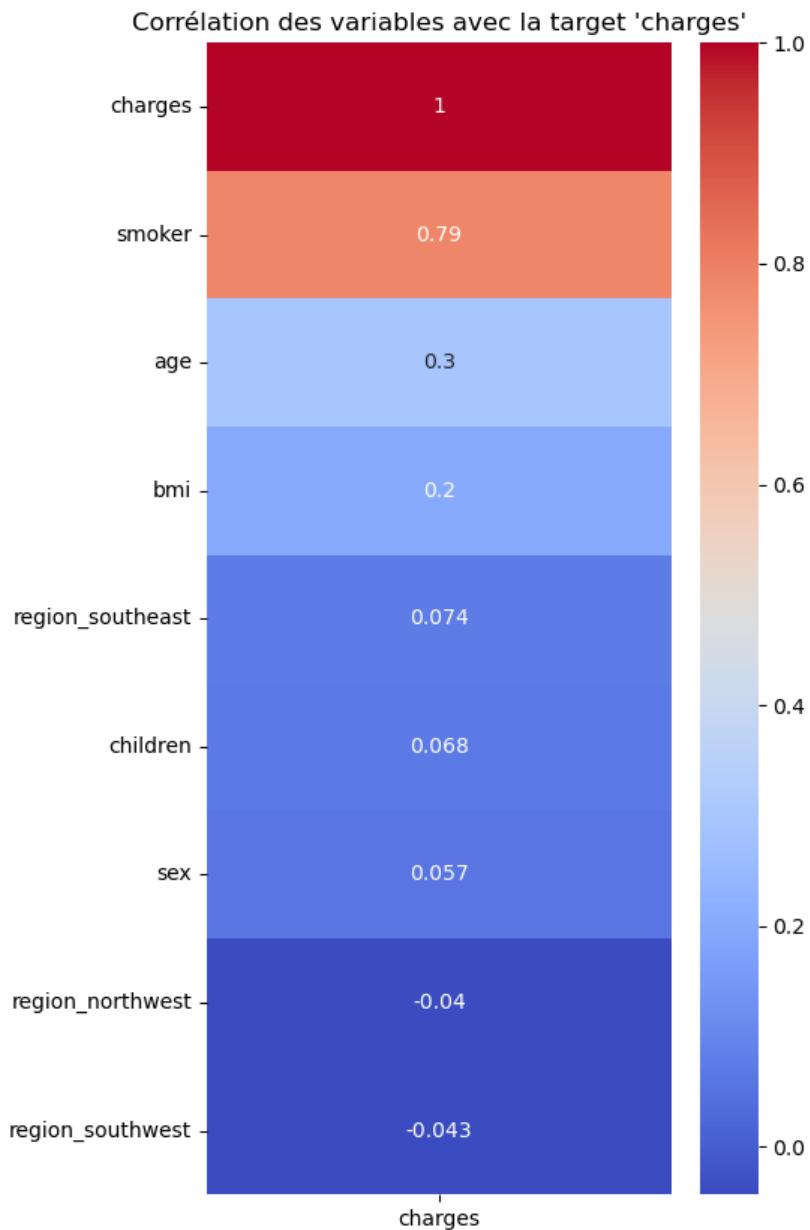
La sélection des features est essentielle pour obtenir un modèle simple, interprétable et performant. J'ai utilisé trois techniques différentes pour faire une bonne sélection.

3.2.1 Corrélation des variables avec la ta et ‘charges’ :



The screenshot shows a Jupyter Notebook cell with three colored dots (red, yellow, green) at the top. The cell contains the following Python code:

```
1 import seaborn as sns
2
3 corr_charges = df3.corr()['charges'].sort_values(ascending=False)
4
5 print("Corrélation:")
6 print(corr_charges)
7
8 # Heatmap de 'charges'
9 plt.figure(figsize=(5,10))
10 sns.heatmap(corr_charges.to_frame(),
11             annot=True,
12             cmap="coolwarm")
13 plt.title("Corrélation des variables avec la target 'charges'")
14 plt.show()
```

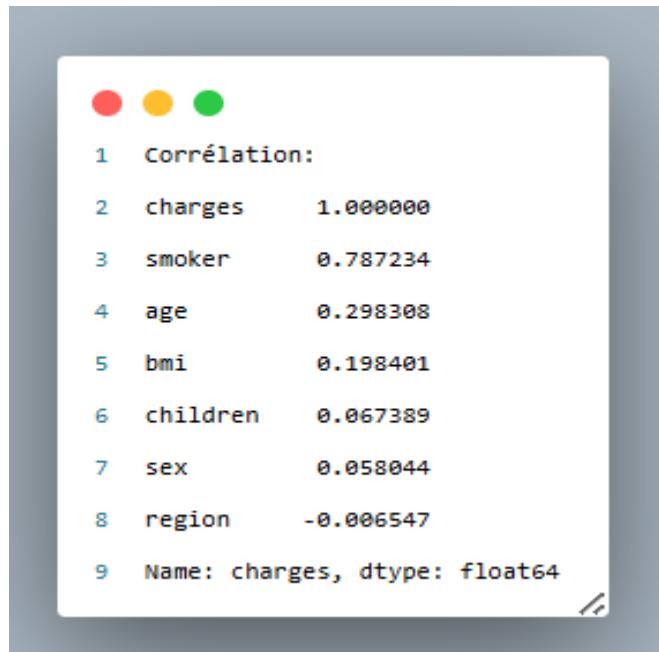


Comme vu précédemment :

- **smoker** → corrélation très forte
- **age** → corrélation élevée
- **bmi** → corrélation significative

Les autres variables comme *children*, *sex* et *region* ont une faible influence.

Les valeurs exactes des corrélations :



The screenshot shows a Jupyter Notebook cell with three colored dots (red, yellow, green) at the top. The code displays a correlation matrix:

```
1 Corrélation:  
2 charges      1.000000  
3 smoker       0.787234  
4 age          0.298308  
5 bmi          0.198401  
6 children     0.067389  
7 sex          0.058044  
8 region      -0.006547  
9 Name: charges, dtype: float64
```

3.2.2 Importance des features (RandomForestRegressor)

Pour valider le choix, nous utilisons un modèle de type Random Forest afin de mesurer l'importance relative de chaque variable.



The screenshot shows a Jupyter Notebook cell with three colored dots (red, yellow, green) at the top. The code uses the RandomForestRegressor to calculate feature importances:

```
1 from sklearn.ensemble import RandomForestRegressor  
2  
3 model_rf = RandomForestRegressor()  
4 model_rf.fit(df3[['age','sex','bmi','children','smoker', 'region']], df3['charges'])  
5  
6 importances = model_rf.feature_importances_  
7 print("Importance des features")  
8 for f, imp in zip(['age','sex','bmi','children','smoker', 'region'], importances):  
9     print(f"{f} : {imp}")
```

Le résultat confirme :

- **smoker** est la feature la plus importante
- **bmi** et **age** suivent juste après



3.2.3 Logique métier :

D'un point de vue métier :

- Un **fumeur** paie naturellement beaucoup plus en assurance santé.
- L'**âge** influence fortement le risque et donc le coût.
- Le **BMI** reflète le niveau de surpoids, lié au risque médical.

3.2.4 Conclusion

Nous retenons logiquement les 3 variables les plus importantes : age, bmi, smoker

3.3. Normalisation / Standardisation

- Pourquoi normaliser ?

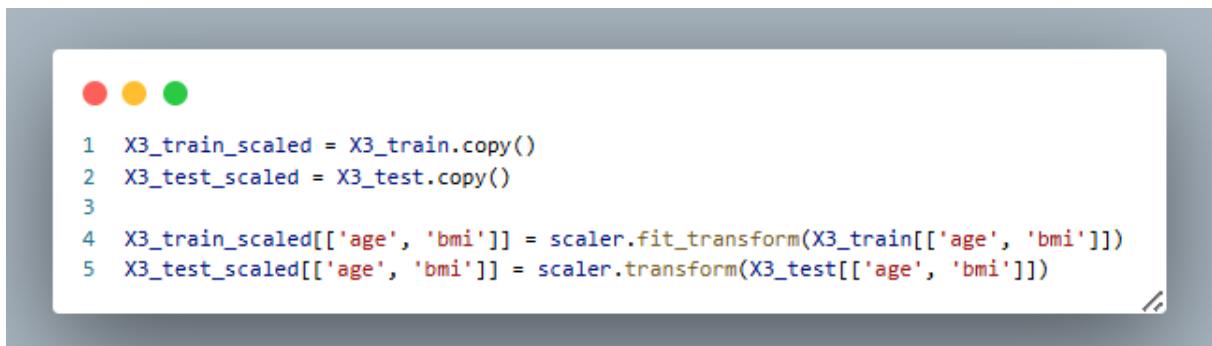
Les trois variables n'ont pas la même échelle :

- age ~ entre 18 et 65
- bmi ~ entre 15 et 50
- smoker ~ 0 ou 1

Une différence d'échelle peut déséquilibrer l'apprentissage et ralentir la convergence du modèle.

La normalisation permet :

- d'améliorer la stabilité du modèle
 - parfois d'améliorer les performances
 - de rendre les coefficients comparables
- Technique choisie : Standardisation



```
1 X3_train_scaled = X3_train.copy()
2 X3_test_scaled = X3_test.copy()
3
4 X3_train_scaled[['age', 'bmi']] = scaler.fit_transform(X3_train[['age', 'bmi']])
5 X3_test_scaled[['age', 'bmi']] = scaler.transform(X3_test[['age', 'bmi']])
```

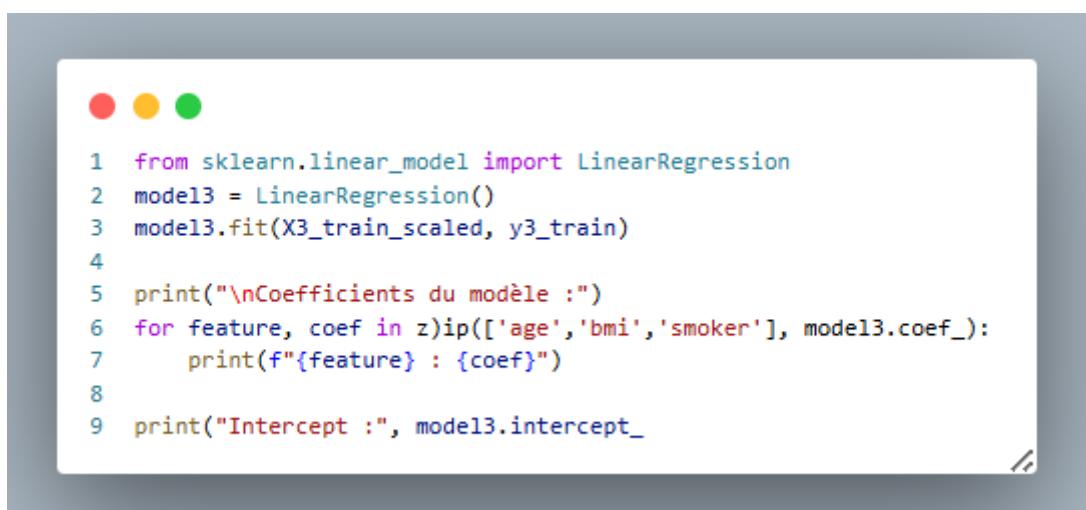
- Pourquoi ce choix ?

La standardisation est recommandée pour la régression linéaire car :

- elle gère bien les distributions non bornées
- elle rend les coefficients plus lisibles
- elle stabilise le calcul des moindres carrés

3.4. Entraînement & Prédiction

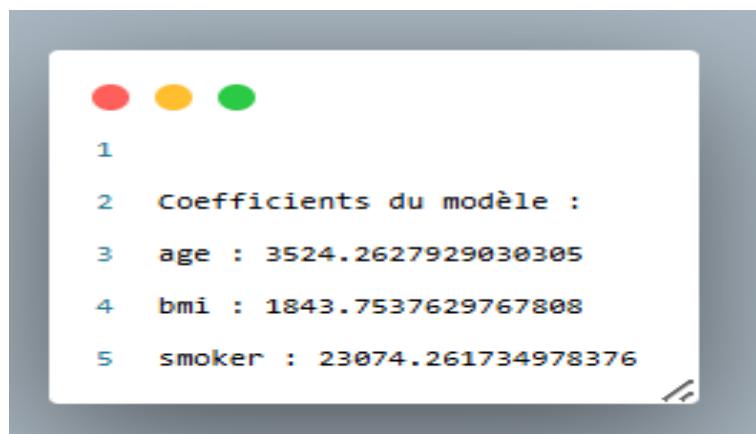
Nous appliquons un modèle de **régression linéaire multiple**.



```
1 from sklearn.linear_model import LinearRegression
2 model3 = LinearRegression()
3 model3.fit(X3_train_scaled, y3_train)
4
5 print("\nCoefficients du modèle :")
6 for feature, coef in zip(['age','bmi','smoker'], model3.coef_):
7     print(f"{feature} : {coef}")
8
9 print("Intercept :", model3.intercept_)
```

De même, le modèle est entraîné sur l'ensemble d'apprentissage (80 % des données).

Puis, les prédictions sont effectuées sur l'ensemble de test.



```
1
2  Coefficients du modèle :
3  age : 3524.2627929030305
4  bmi : 1843.7537629767808
5  smoker : 23074.261734978376
```

3.5. Visualisation

les points suivent la diagonale, cela signifie que les prédictions sont proches des valeurs réelles.

Dans notre cas, les points sont globalement bien alignés mais montrent une dispersion due à la nature complexe du dataset.

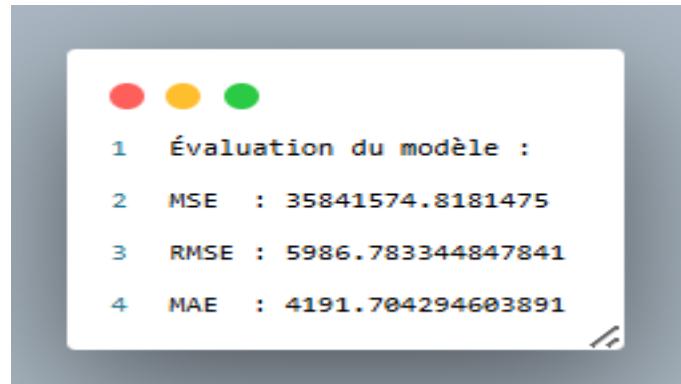


- Résultats observés

Les prédictions montrent que le modèle capture correctement la tendance générale du dataset, même s'il existe une certaine variabilité liée à des facteurs non inclus dans les 3 features choisies.

3.6. Évaluation

Nous utilisons les trois métriques classiques :



Interprétation des résultats

Dans notre cas :

❖ Mean Squared Error (MSE) : 35 841 574.81

- Une valeur élevée est normale dans un dataset où les charges d'assurance peuvent monter à **plus de 60 000**.
- Les grandes erreurs (cas particuliers : fumeurs + BMI élevé) ont augmenté la valeur du MSE.

❖ Root Mean Squared Error (RMSE) : 5 986.78

- Cela signifie qu'en moyenne ton modèle se trompe d'environ $\pm 6\,000$.
- Le modèle généralise bien, mais il garde une erreur notable, logique car les dépenses médicales varient fortement entre individus.

❖ Mean Absolute Error (MAE) : 4 191.70

- Une erreur moyenne d'environ $\pm 4\,200 \$$ est correcte pour un modèle simple linéaire sur ce dataset.
- Le modèle se trompe en moyenne de 4 200 \$, ce qui reste raisonnable vu la variabilité du dataset.

3.7 Conclusion :

- Le modèle est **globalement cohérent**, mais :

- ❖ il reste limité par la **forte variabilité des charges** (surtout les fumeurs et les personnes obèses),

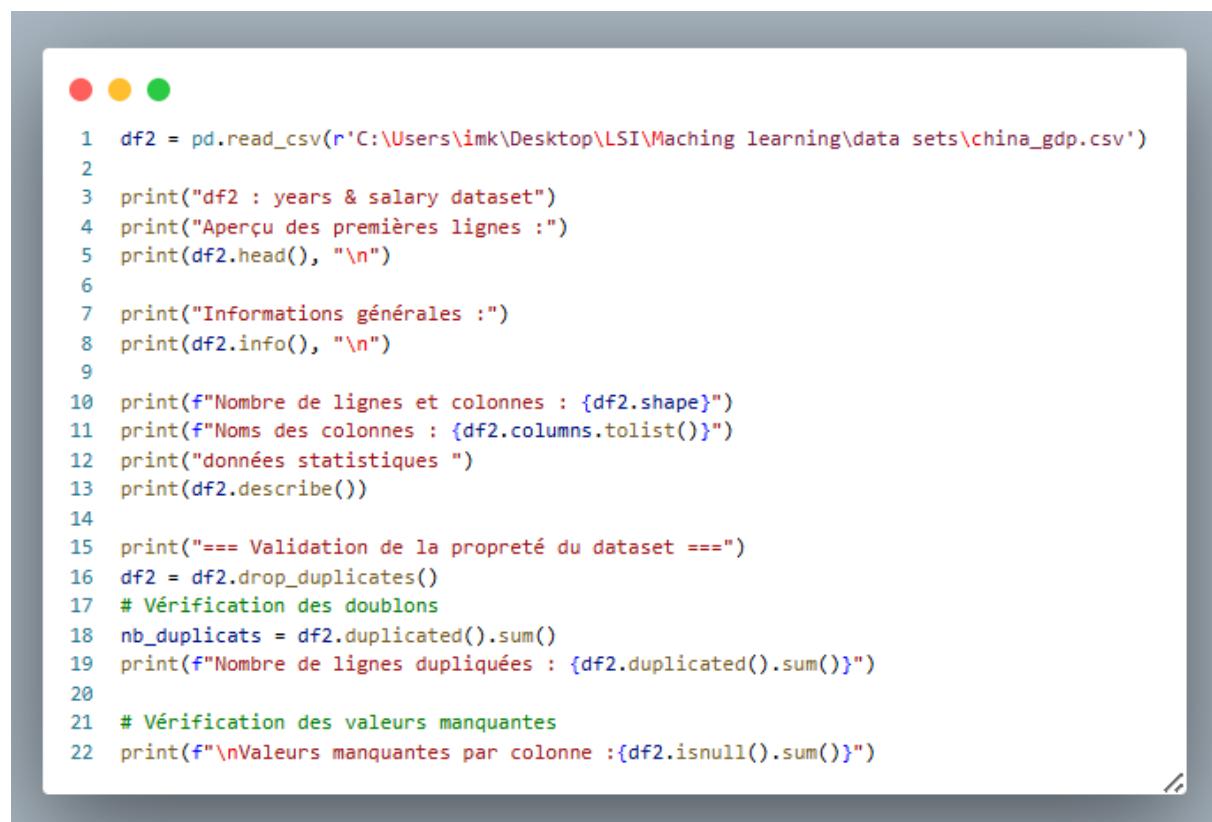
- ❖ les erreurs sont normales pour une régression linéaire simple avec seulement 3 features,

Partie 4 : Régression Linéaire vs Polynomiale (China GDP)

4.1 Préparation des données :

Le dataset China GDP contient l'évolution du PIB de la Chine sur plusieurs décennies.

Visuellement, la croissance du PIB ne suit **pas une droite**, mais plutôt une **courbe en forme de "S"**, typique des phénomènes économiques à long terme (croissance lente au début, accélération, puis stabilisation).



```

1 df2 = pd.read_csv(r'C:\Users\imk\Desktop\LSI\Maching learning\data sets\china_gdp.csv')
2
3 print("df2 : years & salary dataset")
4 print("Aperçu des premières lignes :")
5 print(df2.head(), "\n")
6
7 print("Informations générales :")
8 print(df2.info(), "\n")
9
10 print(f"Nombre de lignes et colonnes : {df2.shape}")
11 print(f"Noms des colonnes : {df2.columns.tolist()}")
12 print("données statistiques ")
13 print(df2.describe())
14
15 print("== Validation de la propreté du dataset ==")
16 df2 = df2.drop_duplicates()
17 # Vérification des doublons
18 nb_duplicats = df2.duplicated().sum()
19 print(f"Nombre de lignes dupliquées : {df2.duplicated().sum()}")
20
21 # Vérification des valeurs manquantes
22 print(f"\nValeurs manquantes par colonne :{df2.isnull().sum()}")

```

Aperçu sur la dataset et ces Informations générales :

```
 1 df2 : years & salary dataset
 2 Aperçu des premières lignes :
 3   Year      Value
 4  0  1960  5.918412e+10
 5  1  1961  4.955705e+10
 6  2  1962  4.668518e+10
 7  3  1963  5.009730e+10
 8  4  1964  5.906225e+10
 9
10 Informations générales :
11 <class 'pandas.core.frame.DataFrame'>
12 RangeIndex: 55 entries, 0 to 54
13 Data columns (total 2 columns):
14 #   Column  Non-Null Count  Dtype  
15 ---  --       --           --      
16 0   Year     55 non-null    int64  
17 1   Value    55 non-null    float64
18 dtypes: float64(1), int64(1)
19 memory usage: 1012.0 bytes
20 None
21
22 Nombre de lignes et colonnes : (55, 2)
23 Noms des colonnes : ['Year', 'Value']
24 données statistiques
25             Year      Value
26 count    55.00000  5.500000e+01
27 mean     1987.00000 1.437042e+12
28 std      16.02082  2.500085e+12
29 min     1960.00000 4.668518e+10
30 25%    1973.50000  1.395123e+11
31 50%    1987.00000  3.074796e+11
32 75%    2000.50000  1.268748e+12
33 max     2014.00000  1.035483e+13
34 === Validation de la propriété du dataset ===
35 Nombre de lignes dupliquées : 0
36
37 Valeurs manquantes par colonne :Year      0
38 Value    0
39 dtype: int64
```

Interprétation du dataset *df2 : china_gdp*

Ce dataset contient 55 observations représentant l'évolution d'un montant économique (appelé *Value*) entre 1960 et 2014. Les données sont propres, sans doublons ni valeurs manquantes. On observe une forte croissance de la

variable *Value* au fil du temps : les valeurs passent d'environ $4,7 \times 10^{10}$ au début des années 1960 à un maximum supérieur à 1×10^{13} en 2014. Les statistiques montrent une distribution très étalée, avec une moyenne très élevée due à l'augmentation progressive de la valeur au fil des décennies. Cette croissance suggère une tendance ascendante marquée, probablement liée à un phénomène économique cumulatif (par exemple : PIB, salaires agrégés, dépenses totales, etc.). L'ensemble du dataset est donc cohérent et prêt à être utilisé pour une analyse temporelle ou pour modéliser une tendance à long terme.

4.2 Modèles entraînés

- **Modèle 1 & 2 : Régression linéaire simple et polynomiale (degré 4)**

L'idée est de vérifier si une relation linéaire est suffisante pour expliquer l'évolution du GDP.

Comme le PIB augmente de manière très rapide sur une courte période, une simple droite ne peut pas suivre la tendance.

Ce modèle permet de représenter des courbes complexes et s'adapte beaucoup mieux à des données non linéaires comme le PIB.

Après transformation, nous entraînons un nouveau modèle de régression linéaire sur ces nouvelles features.

```
● ● ●

1 X2 = df2[['Year']]
2 y2 = df2['Value']
3
4 X2_train, X2_test, y2_train, y2_test = train_test_split(
5     X2, y2, test_size=0.2, random_state=42
6 )
7
8 # model lin
9 lin_model = LinearRegression()
10 lin_model.fit(X2_train, y2_train)
11
12 # model poly
13 poly = PolynomialFeatures(degree=4)
14 X2_poly_train = poly.fit_transform(X2_train)
15 X2_poly_test = poly.transform(X2_test)
16
17 poly_model = LinearRegression()
18 poly_model.fit(X2_poly_train, y2_train)
```

5.3. Résultats et visualisation

Lorsque nous traçons les deux modèles :

- **Résultat du modèle linéaire**

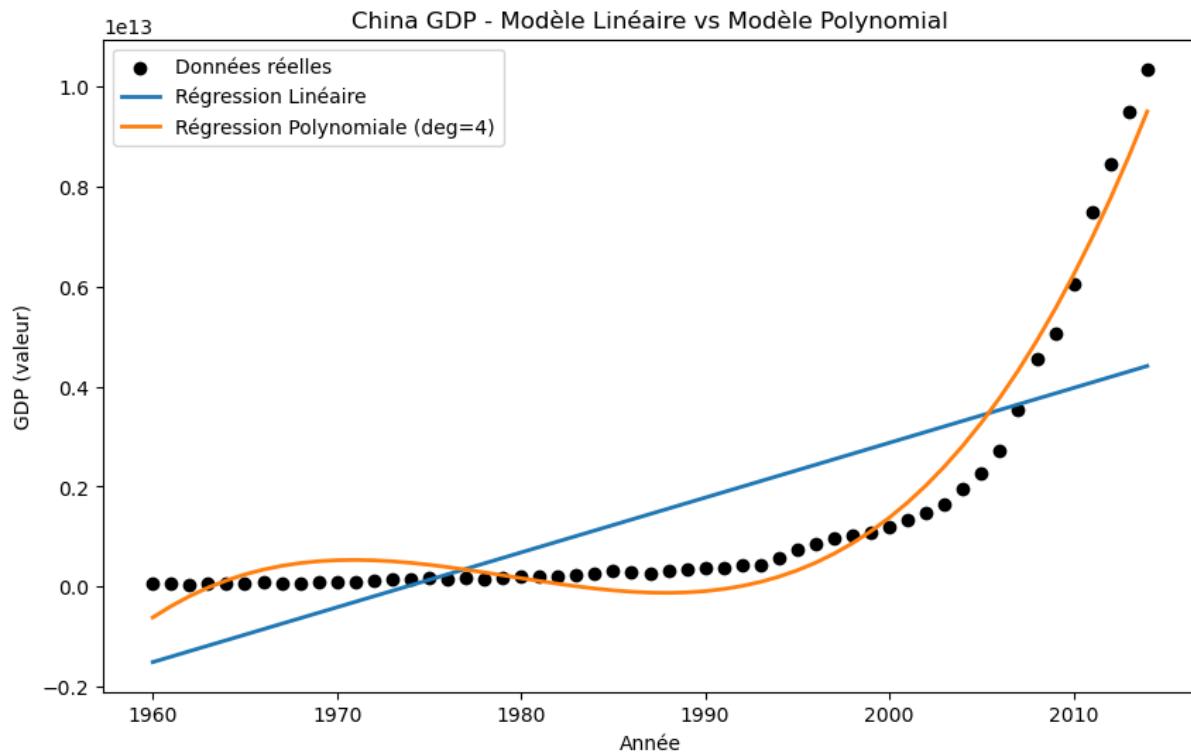
- La droite obtenue est **presque plate**.
- Elle ne suit pas du tout la forme réelle du GDP.
- Elle sous-estime fortement les valeurs récentes du PIB.

Cela confirme que le modèle linéaire est **inadapté** à des données aussi non linéaires.

- **Résultat du modèle polynomial**

- La courbe polynomiale suit beaucoup mieux la tendance réelle.
- Elle reproduit la forme en “S” visible dans les données.
- Les prédictions sont cohérentes sur l’ensemble des années.

Conclusion visuelle :



5.4. Évaluation des deux modèles

Pour comparer objectivement les performances, nous utilisons :

```
1
2 --- Modèle Linéaire ---
3 MSE   : 2.9097229920495426e+24
4 RMSE  : 1705791016522.699
5 MAE   : 1341446144991.7834
6
7 --- Modèle Polynomial (degré 4) ---
8 MSE   : 2.0753412966070593e+23
9 RMSE  : 455559139586.40533
10 MAE  : 413445599274.1387
11
```

Résultats obtenus :

--- *Modèle Linéaire* ---

- MSE très élevé
- RMSE énorme
- MAE très grand

--- *Modèle Polynomial* ---

- MSE **beaucoup plus faible**
- RMSE plus raisonnable
- MAE nettement inférieur au modèle linéaire

Interprétation des résultats

❖ Modèle Linéaire

MSE : 2.90×10^{24} – RMSE : 1.70×10^{12} – MAE : 1.34×10^{12}

Le modèle linéaire échoue totalement : il ne suit pas la forme réelle du PIB chinois, qui est clairement non linéaire (croissance logistique). La droite passe loin des points, d'où des erreurs gigantesques.

❖ Modèle Polynomial (degré 4)

MSE : 2.07×10^{23} – RMSE : 4.55×10^{11} – MAE : 4.13×10^{11}

Les erreurs restent grandes, mais nettement plus faibles que celles du modèle linéaire. Le polynôme reproduit mieux la courbure du PIB, ce qui en fait un modèle beaucoup plus adapté.

Conclusion

Le modèle linéaire est inadapté car il ne peut pas représenter une croissance non linéaire.

Le modèle polynomial de degré 4 suit beaucoup mieux la tendance réelle, bien qu'il conserve de grandes erreurs à cause de l'échelle immense du PIB (ordre de 10^{12}).

- **Linéaire** : très mauvais → forme incompatible.
- **Polynomial 4** : bien meilleur → capture la courbe réelle du PIB.

6. Conclusion générale

Dans ce TP, nous avons appliqué de manière progressive et pratique les principaux concepts liés à la **régression linéaire simple**, **régression linéaire multiple** et **régression polynomiale**, et atelier a permis de comprendre et d'appliquer les principales techniques de régression en Machine Learning, en commençant par l'exploration et la visualisation de plusieurs jeux de données réels. Les analyses graphiques et statistiques ont montré comment les variables évoluent, comment elles interagissent entre elles et quelles sont celles qui influencent le plus la variable cible, notamment dans les cas Expérience-Salaire, Assurance et GDP de la Chine. Ces étapes d'EDA ont servi de base pour choisir les features pertinentes et justifier leur importance dans la construction des modèles.

Ensuite, les différentes formes de régression simple, multiple et polynomiale ont été mises en œuvre grâce à l'API de sklearn, avec entraînement, prédiction, visualisation des résultats et évaluation via MSE, RMSE et MAE. Les comparaisons obtenues ont montré la capacité de chaque modèle à expliquer les données et à prédire de nouvelles valeurs, tout en illustrant les limites de la régression linéaire pour certains phénomènes non linéaires, comme le GDP. Globalement, l'atelier a permis de renforcer la compréhension pratique de la régression et de la préparation de données dans un contexte de Machine Learning.

Fin.