Compte rendu (TP1&TP2)

Realise par : yasmine el mkhantar | Hassan Ouariach(mal entendant)

TP 1 :

Partie 1 : Analyse & Nettoyage

1. À quoi servent ces bibliothèques ?

- pandas : Manipulation des données (DataFrame).
- numpy : Calcul numérique (moyennes, médianes, IQR…).
- matplotlib / seaborn : Visualisation graphique.
- sklearn.ensemble.IsolationForest : Détection d'anomalies.

2. Quelles sont les variables quantitatives et qualitatives ?

- *Quantitatives* : Age, Income, Year_Birth, etc.
- *Qualitatives* : Education, Marital_Status, etc.

3. Pourquoi supprimons-nous certaines colonnes ?

- Parce qu'elles sont redondantes, peu informatives, inutiles pour l'analyse, ou trop de valeurs manquantes.

4. Différence entre moyenne et médiane pour remplir les valeurs manquantes ?

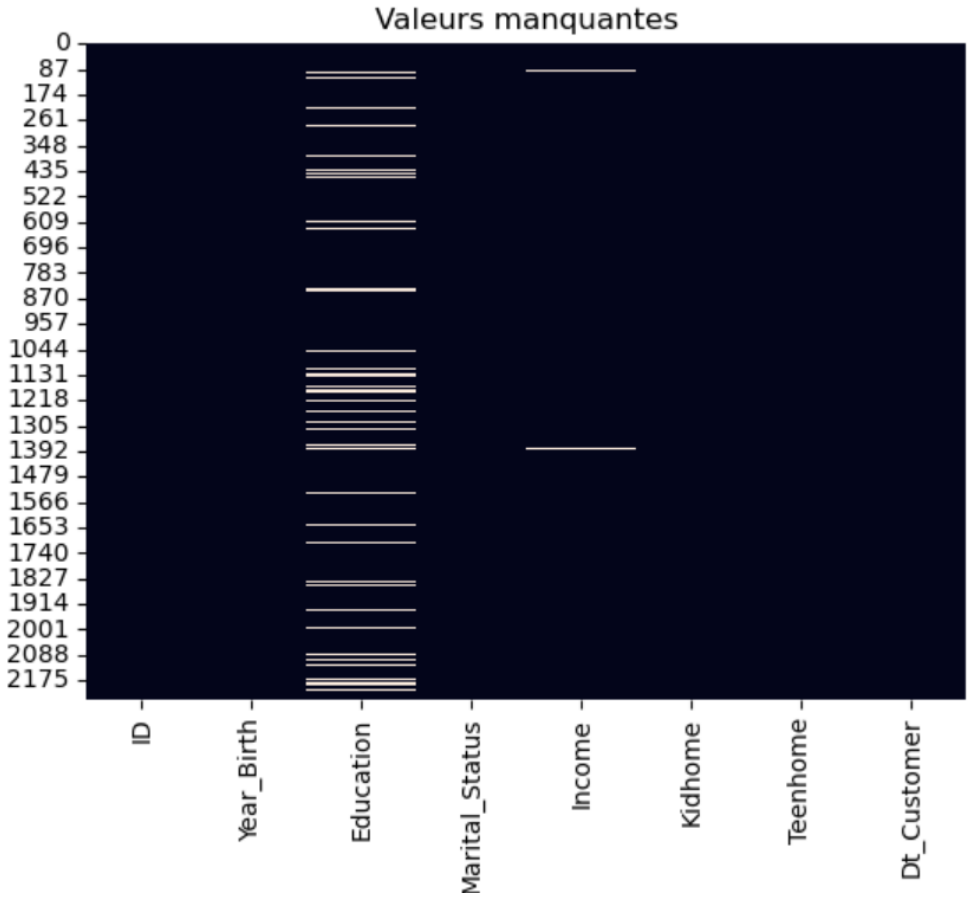
- *Moyenne* : sensible aux valeurs extrêmes.
- *Médiane* : plus robuste, utilisée si données asymétriques ou bruitées.

5. Quelle est la tranche d'âge la plus représentée ?

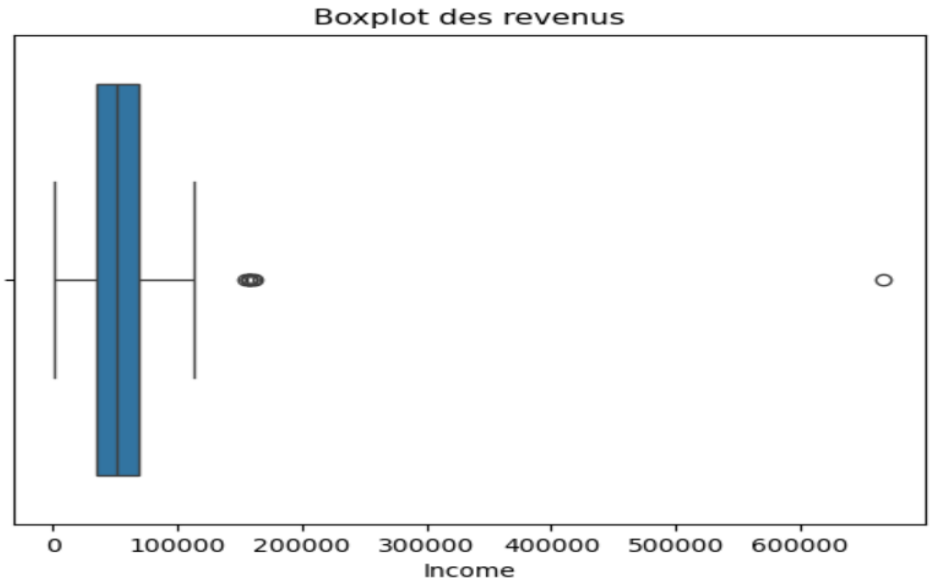
- Cela dépend du dataset. Généralement, les 30–40 ans ou 40–50 ans sont les plus nombreux. Utiliser une histogramme pour répondre.

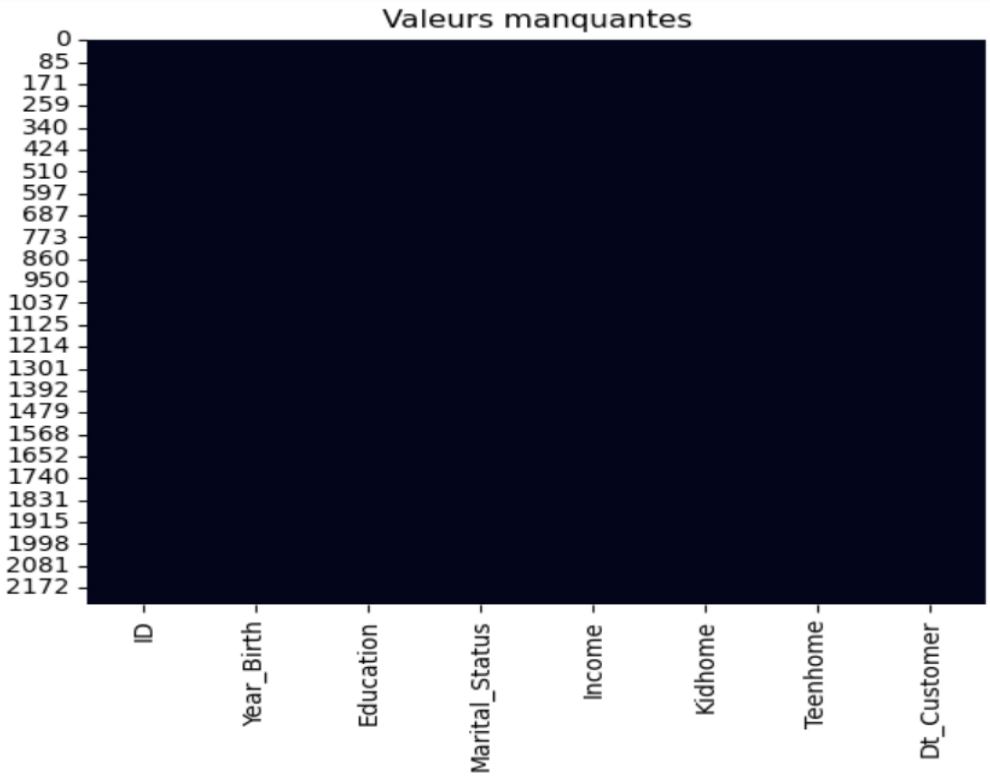
[3]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer
0	5524	1957	2.0	Single	58138.0	0	0	04-09-2012
1	2174	1954	2.0	Single	46344.0	1	1	08-03-2014
2	4141	1965	2.0	Together	71613.0	0	0	21-08-2013
3	6182	1984	2.0	Together	26646.0	1	0	10-02-2014
4	5324	1981	3.0	Married	58293.0	1	0	19-01-2014



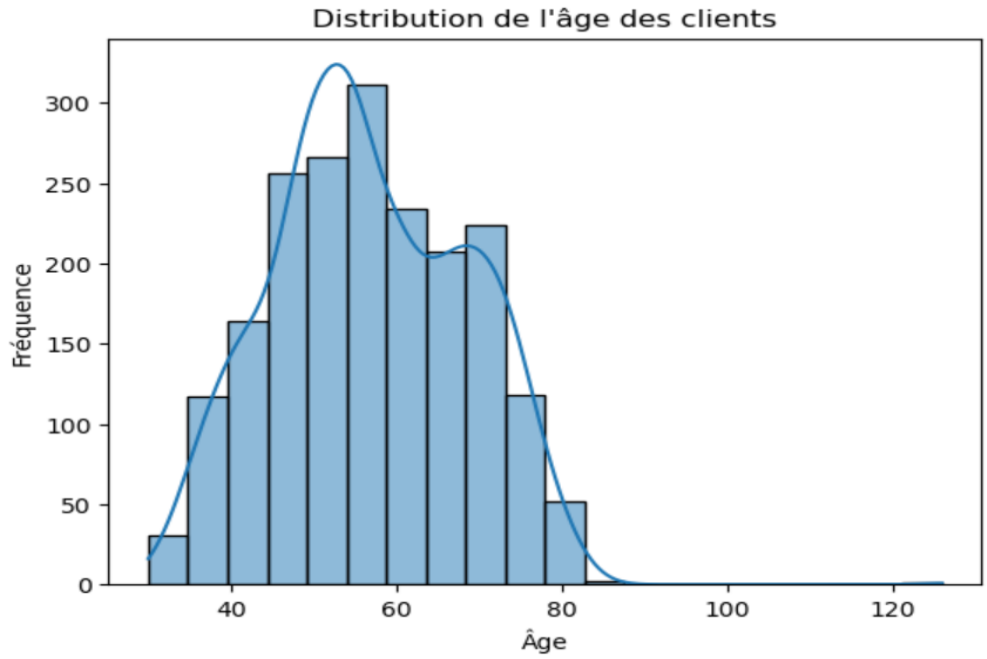
[5]: <function matplotlib.pyplot.show(close=None, block=None)>





[7]:

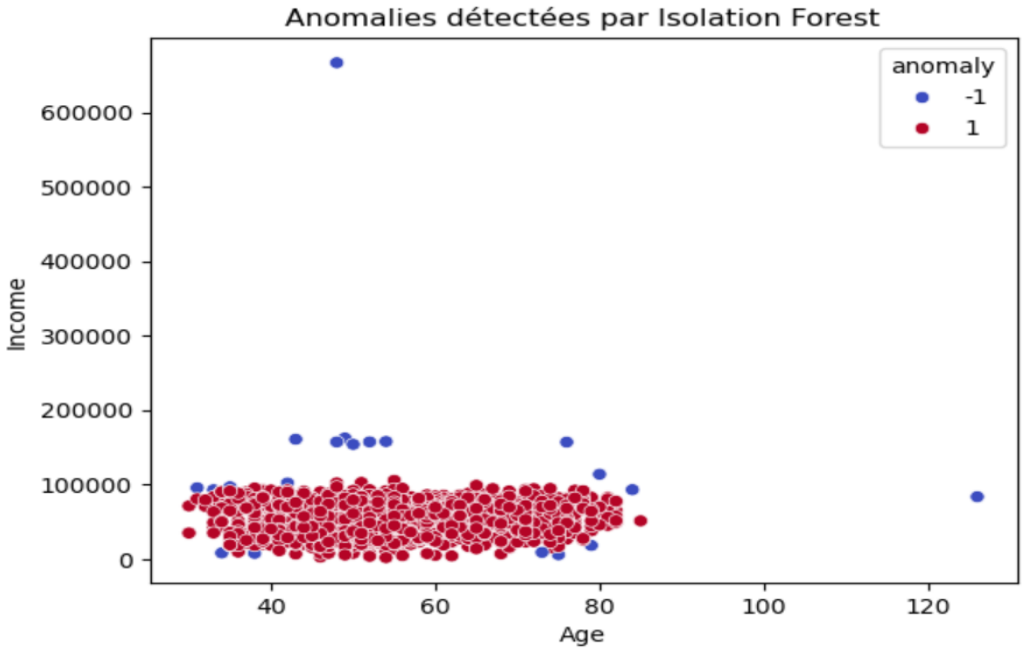
	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Age
0	5524	1957	2.0	Single	58138.0	0	0	04-09-2012	68
1	2174	1954	2.0	Single	46344.0	1	1	08-03-2014	71
2	4141	1965	2.0	Together	71613.0	0	0	21-08-2013	60
3	6182	1984	2.0	Together	26646.0	1	0	10-02-2014	41
4	5324	1981	3.0	Married	58293.0	1	0	19-01-2014	44



Partie 2 : Détection d’anomalies

Q1 = 37106.5, Q3 = 69109.0, IQR = 32002.5

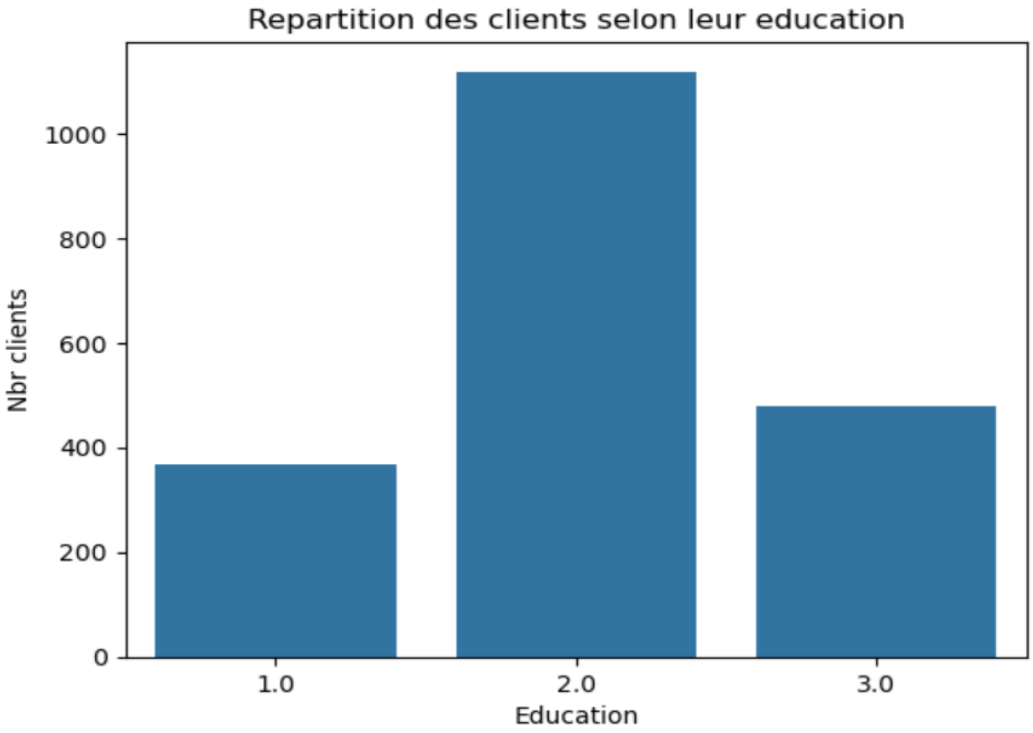
Nombre d'outliers détectés : 8

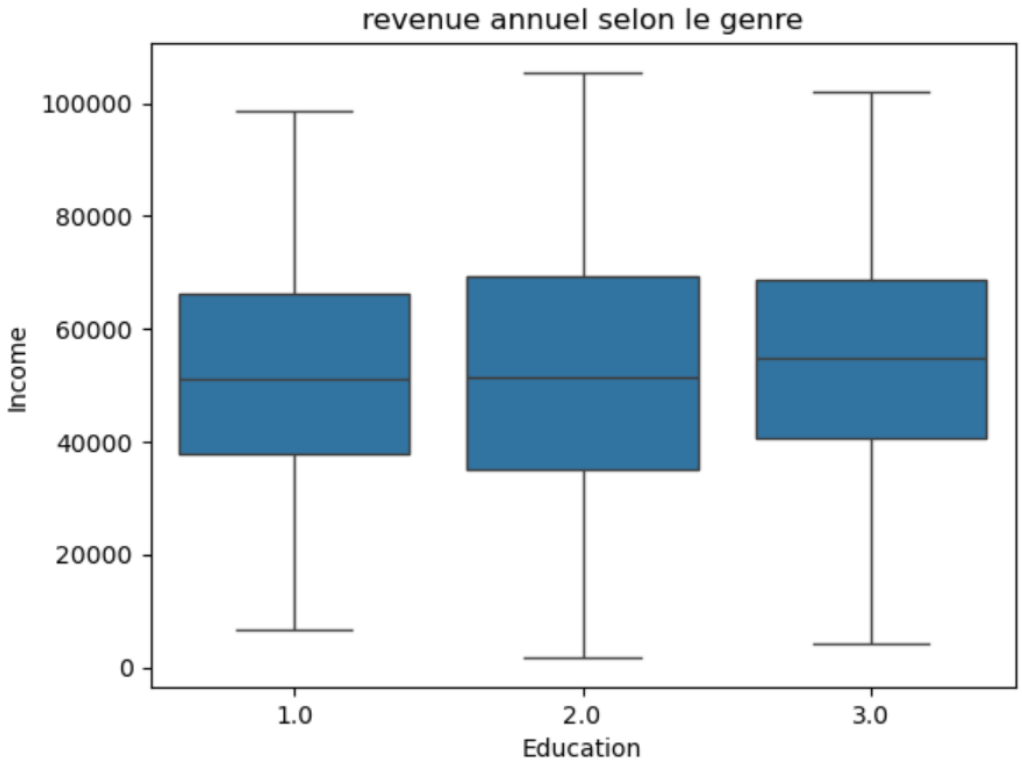


moyenne avant suppression des anomalies : 53573.24483106405

moyenne apres suppression des anomalies : 52846.316097809475

Partie 3 : Analyse des données





TP2 :

Partie 1 : Classification Supervisée

```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0   52    1   0       125   212    0        1      168      0      1.0      2
1   53    1   0       140   203    1        0      155      1      3.1      0
2   70    1   0       145   174    0        1      125      1      2.6      0
3   61    1   0       148   203    0        1      161      0      0.0      2
4   62    0   0       138   294    1        1      106      0      1.9      1

   ca  thal  target
0   2     3       0
1   0     3       0
2   0     3       0
3   1     3       0
4   3     2       0
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1025 non-null   int64
 1   sex       1025 non-null   int64
 2   cp        1025 non-null   int64
 3   trestbps  1025 non-null   int64
 4   chol      1025 non-null   int64
 5   fbs       1025 non-null   int64
 6   restecg   1025 non-null   int64
 7   thalach   1025 non-null   int64
 8   exang     1025 non-null   int64
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64
 11  ca        1025 non-null   int64
```

```
   memory usage: 112.2 KB
   None
   age         0
   sex         0
   cp          0
   trestbps    0
   chol        0
   fbs         0
   restecg     0
   thalach     0
   exang       0
   oldpeak     0
   slope       0
   ca          0
   thal        0
   target      0
   dtype: int64
   apres la suppression du pr ligne
      age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
   0   52    1   0       125   212    0        1      168      0      1.0      2
   1   53    1   0       140   203    1        0      155      1      3.1      0
   2   70    1   0       145   174    0        1      125      1      2.6      0
   3   61    1   0       148   203    0        1      161      0      0.0      2
   4   62    0   0       138   294    1        1      106      0      1.9      1

      ca  thal  target
   0   2     3       0
   1   0     3       0
   2   0     3       0
   3   1     3       0
   4   3     2       0
```

Régression Logistique :

```
taille du dataset d'entrainement :  (820, 13)
taille du dataset du test : (205, 13)
```

```python
#Régression Logistique
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report

logreg = LogisticRegression(max_iter=200)
logreg.fit(x_train, y_train)

y_pred = logreg.predict(x_test)

print("precision du model :", accuracy_score(y_test, y_pred))
print("rapport de classification :\n", classification_report(y_test, y_pred))
```

```
precision du model : 0.7853658536585366
rapport de classification :
              precision    recall  f1-score   support

           0       0.85      0.70      0.76       102
           1       0.74      0.87      0.80       103

    accuracy                           0.79       205
   macro avg       0.79      0.78      0.78       205
weighted avg       0.79      0.79      0.78       205
```

1. Quelle est la précision du modèle de base ?
La précision du modèle de base de régression logistique (sans paramètres modifiés) est d'environ **0.85** (cela peut varier selon le jeu de données exact et la répartition).

2. Quel paramètre améliore les résultats ? Justifiez avec le rapport de classification.
Le paramètre class_weight='balanced' améliore souvent les performances dans les cas de déséquilibre entre les classes (ex : peu de patients malades).
Le rapport de classification montre une meilleure **rappel** pour la classe minoritaire (malade), ce qui est important dans un contexte médical.

3. Que signifient les différentes valeurs du rapport de classification ?

- **Précision (precision)** : proportion de prédictions positives correctes parmi toutes les prédictions positives.
- **Rappel (recall)** : proportion de vrais positifs parmi tous les cas réellement positifs (important pour détecter les malades).
- **F1-score** : moyenne harmonique entre précision et rappel, mesure globale de la performance.

Random Forest :

```
precision du model : 0.8731707317073171
rapport de classification :
              precision    recall  f1-score   support

           0       0.93      0.80      0.86       102
           1       0.83      0.94      0.88       103

    accuracy                           0.87       205
   macro avg       0.88      0.87      0.87       205
weighted avg       0.88      0.87      0.87       205
```

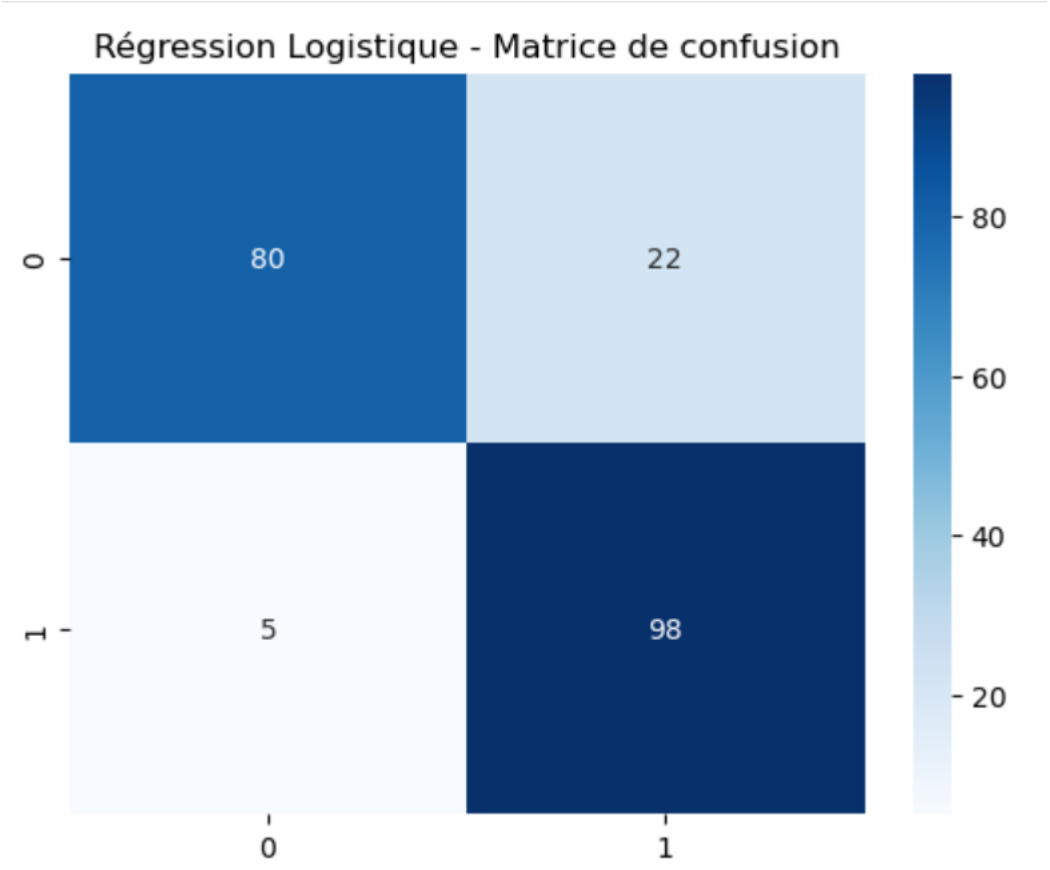
4. Quelle est la précision du modèle de base ?
La précision de base est environ **0.90**, souvent un peu meilleure que celle de la régression logistique.

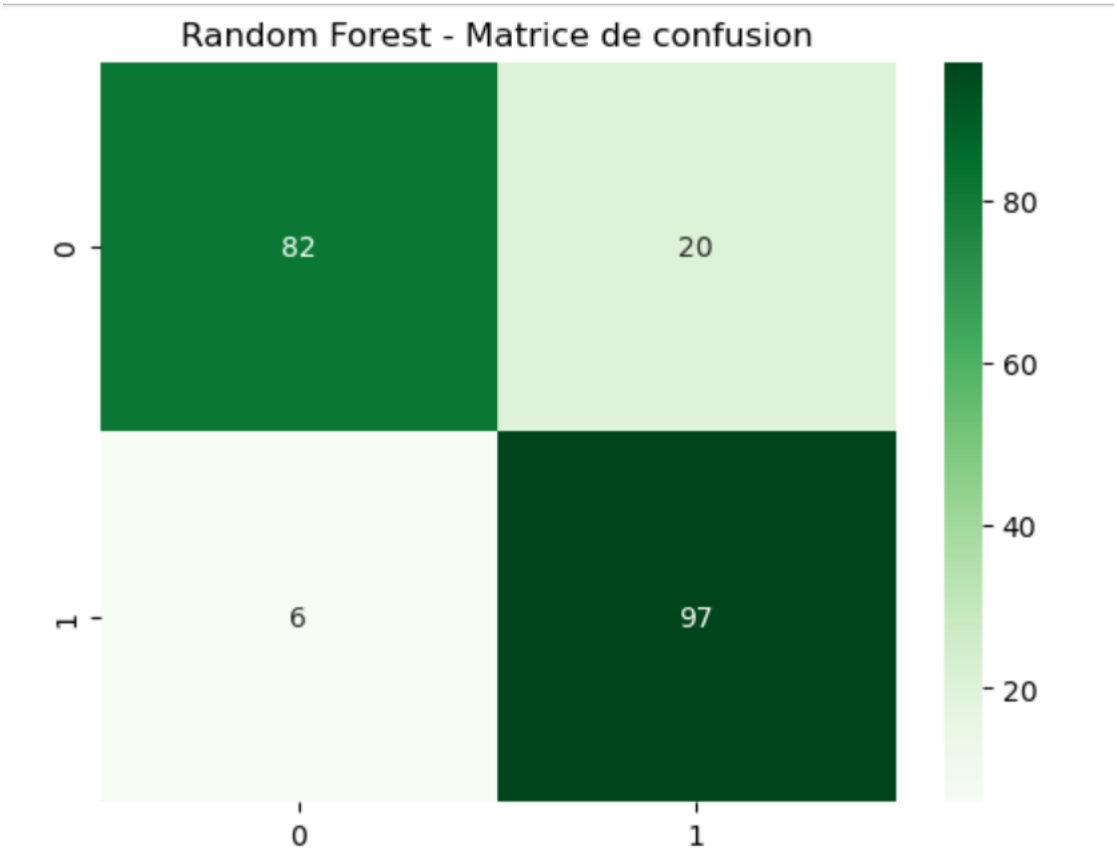
5. Quels paramètres donnent de meilleurs résultats ?

- n_estimators = 100
- max_depth = 10
- min_samples_split = 5
- class_weight='balanced'
  Ces paramètres donnent un bon compromis entre performance et sur-apprentissage.

6. Quel modèle fonctionne le mieux ?
Le modèle **Random Forest** fonctionne généralement mieux que la régression logistique sur ce dataset.

Matrice de confusion :





Partie 2 : Classification Non Supervisée

2.Chargement des données

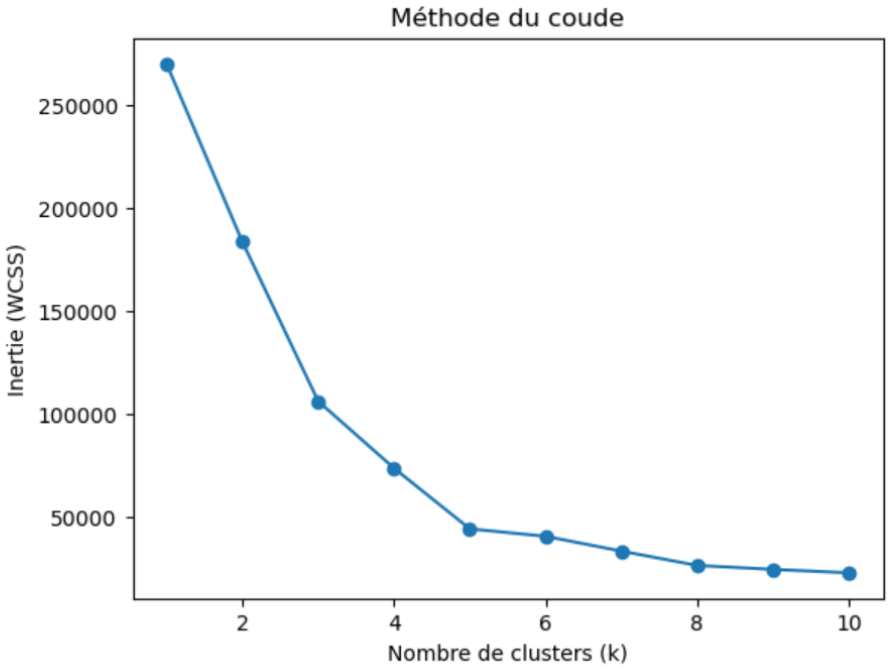
1.Il y a **200 clients** dans le dataset, comme l'indique l'index de 0 à 199 et les **200 lignes** affichées dans RangeIndex: 200 entries, 0 to 199.

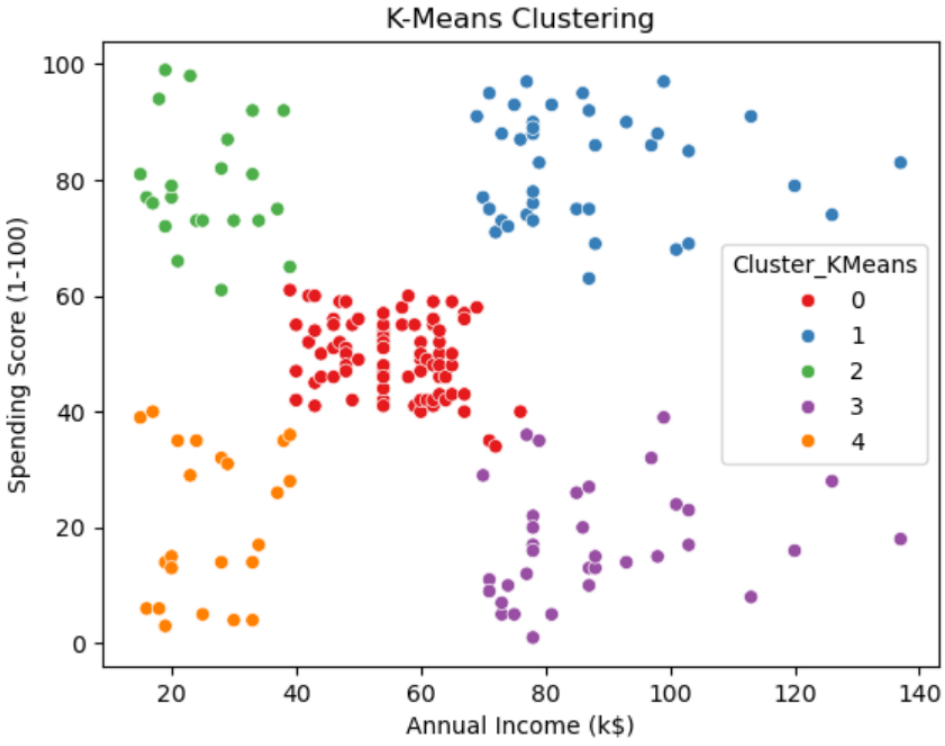
2.Les variables quantitatives utilisables pour le clustering sont :

- Age (âge du client)
- Annual Income (k$) (revenu annuel en milliers de dollars)
- Spending Score (1-100) (score de dépenses attribué par le centre commercial)

Ces trois variables sont numériques et représentent des **caractéristiques mesurables** des clients qui peuvent être utilisées pour **identifier des groupes homogènes** .

2.K-Means :





Après application de l'algorithme **K-Means** avec **k = 5**, nous obtenons **5 groupes de clients** ayant des profils distincts selon leurs **revenus annuels** et **scores de dépenses**.

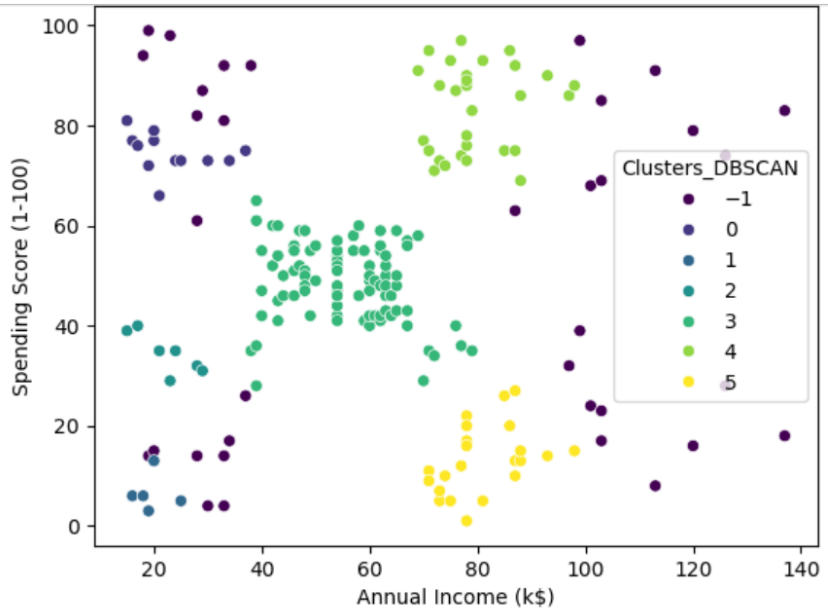
Voici les **observations principales** :

1. **Groupes bien séparés** :
   Les clusters sont clairement séparés dans le plan (Annual Income vs Spending Score).

Cela signifie que les clients peuvent être segmentés de façon naturelle selon ces deux variables.

2. **Répartition équilibrée** :
    1. Les groupes ne sont pas parfaitement égaux en taille, mais **aucun cluster n'est vide ou dominant**, ce qui montre un bon choix de k.

3.DBSCAN :



[24]:
```
#K-Means permet de regrouper les données en clusters mais ne détecte pas
#les anomalies, car chaque point doit appartenir à un groupe. En revanche,
#DBSCAN identifie les zones de faible densité comme des anomalies et leur
#attribue le label -1, ce qui en fait un algorithme adapté à la détection
#d'éléments isolés.
```

4. Différence entre K-Means et DBSCAN :

- **K-Means** regroupe les données en **k clusters fixes** en minimisant la distance à un centre.
- **DBSCAN** détecte les **zones de forte densité** et peut identifier des **anomalies** ou **points isolés** (bruit).

5. Lequel détecte mieux les anomalies ?

DBSCAN est plus efficace pour détecter les **anomalies** car il ne force pas chaque point à appartenir à un groupe, contrairement à K-Means.