

RAPPORT D'ÉTUDE :

PREDICTING CANCER SEVERITY FROM PATIENT DATA

Table des matières :

Introduction.....	3
I. Source des données : enquête HINTS 7 (2024).....	3
II. Analyse exploratoire des données (EDA).....	4
III. Dictionary of Question.....	4
IV. Crédit à la création d'un nouveau DataFrame.....	4
A. Remplacement des valeurs « 2 » par « 0 ».....	4
B. Variable Person W : pondération par individu.....	4
C. Analyse des valeurs manquantes.....	5
D. Crédit à la création de nouvelles variables catégorielles.....	5
1. SleepCategory.....	6
2. DrinkFreqCategory.....	7
3. TimesSunburned_Frequency.....	8
4. AgeGroup.....	10
5. CutSkipMeals2_cat.....	11
V. Test Chi².....	13
A. Pourquoi utiliser un test du Chi ² (Chi-deux) ?.....	13
B. Le test du Chi ² : Déterminer la dépendance entre variables.....	13
C. Top 5 des cancers les plus fréquents dans notre échantillon.....	14
D. Sélection des variables liées aux habitudes de vie.....	14
1. Les Variables Binaire.....	16
2. Les Variables Catégorielles OrdinalesRésultat (CaSkin):.....	18
3. Les Variables Numériques Continues.....	20
Conclusion de l'analyse exploratoire (focus sur CaSkin).....	21
VI. Modélisation : Vers une détection fiable du risque de cancer.....	22
A. Quel modèle prédit le mieux ? Test avec PyCarret.....	22
1. PyCarret.....	22
B. Mise en pratique des résultats obtenus : Évaluation de notre modèle.....	22
1. Random Forest.....	22
2. Analyse des performances du modèle Random Forest pour la détection du cancer.....	24
C. Equilibrage Manuellement de l'échantillonnage.....	25
1. Équilibrage avec Undersampling.....	25
D. Cross Validation : Vers une validation robuste.....	27
1. Évaluation avec validation croisée et undersampling.....	27
2. Évaluation avec Validation Croisée et SMOTE.....	28
3. Comparaison des différentes approches d'équilibrage et d'évaluation.....	29
a. Modèle sans équilibrage (dataset déséquilibré).....	29
b. Équilibrage par undersampling.....	29
c. Équilibrage par SMOTE (oversampling synthétique).....	30
d. Validation croisée + SMOTE.....	30
E. Régression Logistique : Comparaison avec un modèle linéaire.....	31
1. Modèle : Régression Logistique (avec undersampling).....	31

2. Modèle : Régression Logistique (avec SMOTE).....	31
F. Analyse comparative des modèles de classification.....	32
1. Déséquilibre des classes et stratégies de correction.....	32
2. Méthodologie d'évaluation.....	32
3. Résultats clés.....	32
G. Modèle Random Forest avec undersampling et validation croisée.....	33

Predicting Cancer Severity from Patient Data

Introduction

Le cancer demeure l'une des principales causes de mortalité à l'échelle mondiale¹, soit 1 décès sur 6. Si les facteurs génétiques y contribuent, de nombreux types de cancers sont également liés à des déterminants comportementaux et environnementaux, tels que le régime alimentaire, le tabagisme, le niveau de stress ou encore l'accès aux soins de santé.

Ce projet s'inscrit dans une approche de prévention, en étudiant la possibilité de prédire le risque de cancer à partir des habitudes de vie déclarées par les patients. Pour cela, nous exploitons un jeu de données riche, incluant des variables de santé, de mode de vie, de statut socio-économique et de conditions médicales.

Notre démarche se décline en plusieurs étapes :

- Nettoyage et préparation des données,
- Sélection des variables les plus pertinentes,
- Construction de modèles prédictifs (Random Forest, Régression Logistique),
- Ré-échantillonnage manuel et rééquilibrage du jeu de données à l'aide de la méthode SMOTE,
- Évaluation des performances à travers une validation croisée rigoureuse.

L'objectif final est double : identifier les facteurs les plus associés au risque de cancer, et évaluer la pertinence d'une modélisation prédictive basée uniquement sur des données auto-déclarées.

I. Source des données : enquête HINTS 7 (2024)

Les analyses présentées dans cette étude reposent sur les données de la Health Information National Trends Survey – HINTS 7 (2024), une enquête nationale réalisée par le National Cancer Institute (NCI) aux États-Unis. Cette enquête, administrée entre mars et septembre 2024, comprend 77 questions et vise à recueillir des informations sur les comportements de santé, les habitudes de vie, ainsi que l'accès à l'information médicale au sein de la population adulte américaine.

L'enquête a été menée selon une méthodologie d'échantillonnage aléatoire en deux étapes, incluant d'abord une sélection d'adresses, suivie d'un tirage aléatoire d'un individu par foyer. Les participants pouvaient répondre par courrier ou en ligne, avec une incitation financière de 2 \$ (en accompagnement du questionnaire) et 10 \$ à la complétion.

Des efforts spécifiques ont été faits pour améliorer la qualité et la représentativité des données, notamment : un engagement de sincérité demandé aux répondants concernant la véracité de leurs réponses ; une incitation supplémentaire de 10 \$ pour les foyers situés dans des zones à forte concentration de populations minoritaires.

La taille initiale de l'échantillon comptait 7278 participants.

¹ Organisation Mondiale de la santé (OMS), 2022

Avant de construire un modèle prédictif, il est essentiel de se familiariser avec la structure et le contenu des données. C'est l'objet de l'analyse exploratoire suivante, qui permet de repérer les tendances générales, les valeurs manquantes et les variables les plus pertinentes.

II. Analyse exploratoire des données (EDA)

Afin de prédire la sévérité du cancer, nous avons commencé par explorer en détail la structure et la qualité du jeu de données. Cette phase a permis d'identifier les types de variables disponibles (numériques, ordinaires, catégorielles), d'évaluer la présence de valeurs manquantes et de produire des statistiques descriptives de base.

Un dictionnaire des variables a été construit à partir du questionnaire initial, facilitant l'interprétation des colonnes, la détection des redondances et la compréhension des réponses possibles. Ce travail de documentation s'est avéré crucial pour guider le nettoyage, l'encodage et la transformation des données.

L'analyse exploratoire qui en découle a mis en évidence plusieurs variables associées de manière significative au fait d'avoir eu un cancer, notamment des facteurs liés à l'accès aux soins, aux antécédents familiaux et à certaines habitudes de vie.

Forts de ces constats, nous avons constitué un jeu de données propre et cohérent, intégrant uniquement les variables les plus pertinentes. Nous allons à présent passer à la phase de modélisation, en testant différentes approches de classification afin d'évaluer leur capacité à prédire la survenue d'un cancer à partir des données déclaratives.

III. Dictionary of Question

Nous avons créé un dictionnaire de données regroupant l'ensemble des variables issues du questionnaire initial, accompagné de leurs descriptions. Cet outil facilite la compréhension de la signification de chaque colonne et oriente efficacement les étapes de nettoyage et de préparation des données. Il constitue une étape essentielle pour garantir la cohérence, la clarté et la reproductibilité de notre analyse.

(📎 Voir document en annexe : Bibliothèque des variables)

À noter : les lettres allant de C à R correspondent à la numérotation des questions telles qu'elles figurent dans le questionnaire d'origine.

IV. Création d'un nouveau DataFrame

Avant d'entraîner nos modèles, nous devons effectuer plusieurs opérations de transformation et de nettoyage afin de garantir la qualité et la cohérence des données utilisées.

A. Remplacement des valeurs « 2 » par « 0 »

Dans certaines variables, la valeur 2 correspond au choix « No ». Afin de simplifier l'analyse, nous avons choisi de la convertir en 0, indiquant une réponse négative. Cela évite toute ambiguïté lors de la modélisation et permet d'avoir des variables binaires 0/1 pour Oui/Non.

B. Variable Person W : pondération par individu

Dans notre jeu de données issu de l'enquête HINTS7², chaque **ligne représente un individu**, mais certains répondants sont statistiquement représentatifs d'un plus grand nombre de personnes dans la population.

La variable **Person W (person weight)** est un **poids d'échantillonnage** attribué à chaque personne. Elle permet de compenser les groupes sous-représentés dans l'échantillon en leur attribuant un poids plus important. Ainsi :

- Les résultats deviennent **plus représentatifs** de la population générale.
- Chaque groupe (âge, sexe, revenu...) est **équitablement pris en compte** dans les analyses descriptives ou les modèles prédictifs.

NB : Non - utilisation de la variable Person Weight (PERSON_FINWT0) dans la modélisation

Initialement, la variable PERSON_FINWT0, représentant un poids d'échantillonnage individuel, avait été envisagée pour corriger les déséquilibres de notre échantillon et rendre le modèle plus représentatif de la population.

Après plusieurs expérimentations, nous avons constaté que son intégration dans l'entraînement complexifie l'apprentissage, notamment en impactant négativement le rappel de la classe à risque. Par ailleurs, les techniques de rééchantillonnage, notamment SMOTE couplé à un sous-échantillonnage manuel, se sont révélées plus efficaces pour gérer ce déséquilibre.

Par conséquent, la variable PERSON_FINWT0 n'a pas été intégrée dans le modèle final. Cette décision simplifie le pipeline tout en maintenant une bonne performance prédictive.

La variable est néanmoins conservée dans la documentation, afin de rappeler son rôle et sa pertinence dans le cadre de l'enquête HINTS7.

C. Analyse des valeurs manquantes

Une évaluation approfondie de la **qualité des données** a été réalisée afin d'identifier les colonnes comportant un nombre élevé de valeurs manquantes.

Nous avons calculé la **proportion de valeurs manquantes (NaN) pour chaque colonne**, puis filtré celles où **plus de 50 % des valeurs sont absentes**. Ces variables ont été isolées dans un DataFrame distinct nommé `df_filtered`.

Cela nous permet :

- **d'anticiper les variables peu exploitables** pour la modélisation,
- de décider ultérieurement si elles doivent être **exclues, imputées ou analysées séparément**.

D. Création de nouvelles variables catégorielles

Certaines variables continues, telles que :

² HINTS 7 (2024) (*Health Information National Trends Survey*), réalisée par le National Cancer Institute (NCI) aux États-Unis. <https://hints.cancer.gov/>

- SleepWeekdayHr (heures de sommeil en semaine),
 - DrinkDaysPerMonth (jours de consommation d'alcool par mois),
 - TimesSunburned (nombre de coups de soleil),
- sont difficilement interprétables directement. Pour faciliter leur analyse, **nous les avons transformées en variables catégorielles**, en les regroupant selon des seuils cliniquement ou logiquement pertinents.

1. SleepCategory

À partir de la variable SleepWeekdayHr, nous avons créé une nouvelle variable SleepCategory selon la logique suivante :

```

import pandas as pd
import plotly.express as px

# ✅ Nettoyer et convertir la colonne existante
df_final['SleepWeekdayHr'] = pd.to_numeric(df_final['SleepWeekdayHr'], errors='coerce')

# ✅ Classification
def classify_sleep(hours):
    if pd.isna(hours) or hours < 0:
        return 'Inconnu'
    elif hours <= 5:
        return 'Très court à court'
    elif hours in [6, 7, 8]:
        return 'Normal'
    elif hours in [9, 10]:
        return 'Long'
    elif hours >= 11:
        return 'Très long'
    else:
        return 'Inconnu'

df_final['SleepCategory'] = df_final['SleepWeekdayHr'].apply(classify_sleep)
df_final['SleepCategory'] = pd.Categorical(df_final['SleepCategory'],
                                           categories=['Très court à court', 'Normal', 'Long', 'Très long', 'Inconnu'],
                                           ordered=True)

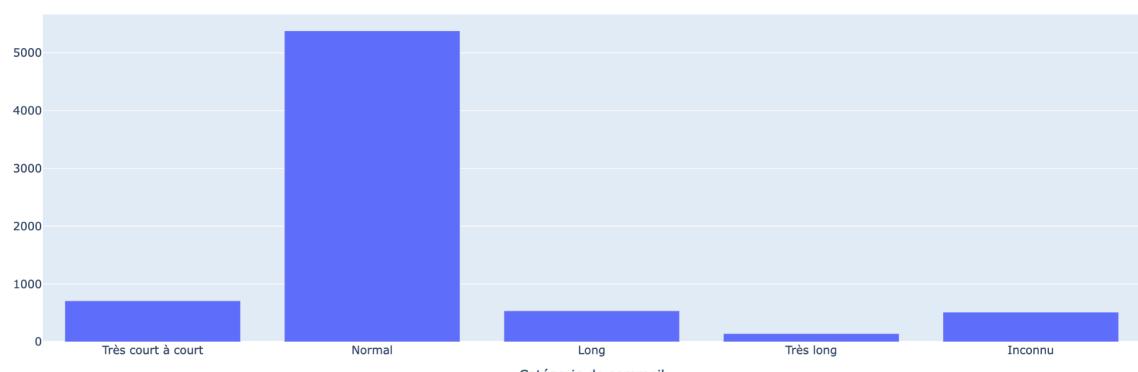
# ✅ Comptage pour le graphique
sleep_counts = df_final['SleepCategory'].value_counts().reindex(df_final['SleepCategory'].cat.categories).reset_index()
sleep_counts.columns = ['SleepCategory', 'count']

# ✅ Graphique
fig = px.bar(
    sleep_counts,
    x='SleepCategory',
    y='count',
    text='count',
    labels={'SleepCategory': 'Catégorie de sommeil', 'count': 'Nombre de personnes'},
    title='Distribution des catégories de sommeil (jours de semaine)'
)

fig.update_traces(textposition='outside')
fig.update_layout(xaxis_title='Catégorie de sommeil', yaxis_title='Nombre de répondants')
fig.show()

```

Distribution des catégories de sommeil (jours de semaine)



Analyse des habitudes de sommeil

L'analyse de la variable SleepWeekdayHr, regroupée en catégories, montre que **la majorité des participants dorment entre 6 et 8 heures en semaine**, ce qui correspond à une durée de sommeil dite "normale". Cette catégorie représente de loin **la plus grande part des répondants**, avec plus de 5 000 personnes, soit plus de 68%.

Les autres catégories — notamment "**Très court à court**" (≤ 5 h) et "**Très long**" (≥ 11 h) — sont **moins représentées**, mais méritent une attention particulière. Ces durées extrêmes sont souvent associées à des effets délétères sur la santé, et pourraient potentiellement être corrélées à une prévalence plus élevée de certaines maladies, dont le cancer³.

Cette classification préparée permettra **d'examiner plus facilement les corrélations** entre le sommeil et différents types de cancers dans les analyses suivantes. En effet, **une durée de sommeil inhabituelle pourrait constituer un indicateur de risque**, ou être le reflet d'un mode de vie globalement défavorable à la santé.

2. DrinkFreqCategory

```
import pandas as pd
import plotly.express as px

# ✅ Fonction de catégorisation
def categorize_drink_days(days):
    if pd.isna(days) or days < 0:
        return 'Inconnu'
    elif days == 0:
        return 'Aucun'
    elif 1 <= days <= 3:
        return 'Rare'
    elif 4 <= days <= 9:
        return 'Occasionnel'
    elif 10 <= days <= 19:
        return 'Régulier'
    elif 20 <= days <= 29:
        return 'Fréquent'
    elif days == 30:
        return 'Quotidien'
    else:
        return 'Inconnu'

# ✅ Appliquer la fonction
df_final['DrinkFreqCategory'] = df_final['DrinkDaysPerMonth'].apply(categorize_drink_days)

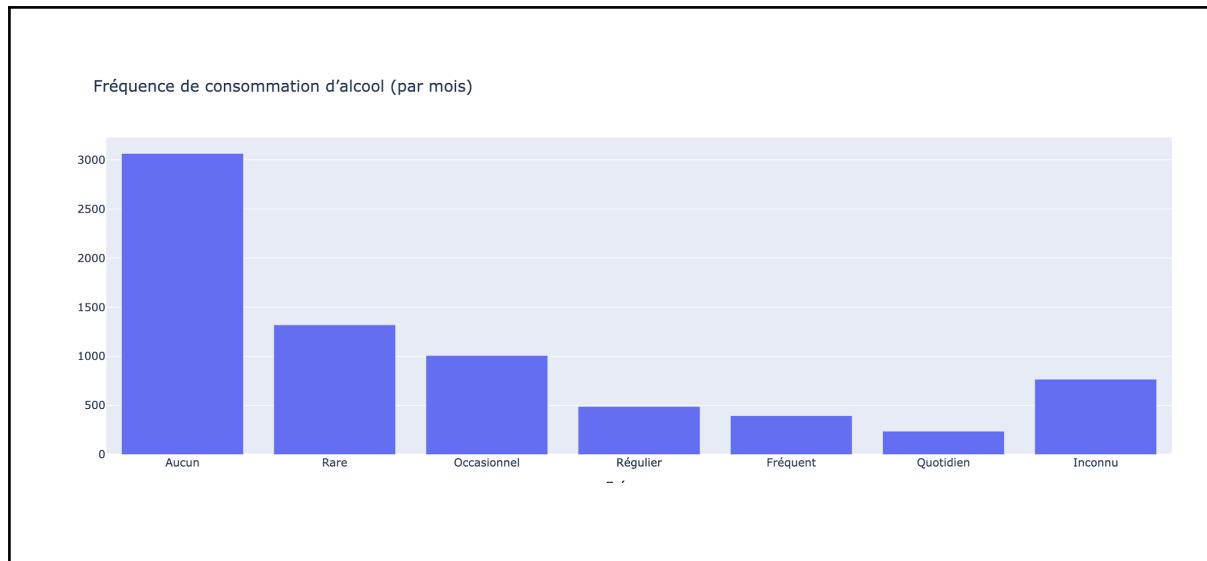
# ✅ Ordre des catégories
categorie_order = ['Aucun', 'Rare', 'Occasionnel', 'Régulier', 'Fréquent', 'Quotidien', 'Inconnu']
df_final['DrinkFreqCategory'] = pd.Categorical(df_final['DrinkFreqCategory'], categories=categorie_order, ordered=True)

# ✅ Compter les valeurs
drink_counts = df_final['DrinkFreqCategory'].value_counts().reindex(categorie_order).reset_index()
drink_counts.columns = ['DrinkFreqCategory', 'count']
```

```
# ✅ Affichage graphique
fig = px.bar(
    drink_counts,
    x='DrinkFreqCategory',
    y='count',
    text='count',
    labels={'DrinkFreqCategory': 'Catégorie de consommation', 'count': 'Nombre de personnes'},
    title='Fréquence de consommation d\'alcool (par mois)'
)

fig.update_traces(textposition='outside')
fig.update_layout(yaxis_title='Nombre de répondants', xaxis_title='Fréquence', uniformtext_minsize=8)
fig.show()
```

³Ma QQ, Yao Q, Lin L, Chen GC, Yu JB. Sleep duration and total cancer mortality: a meta-analysis of prospective studies. *Sleep Med*. 2016 Nov-Dec;27-28:39-44. doi: 10.1016/j.sleep.2016.06.036. Epub 2016 Nov 1. PMID: 27938917.



Analyse de la fréquence de consommation d'alcool

La variable DrinkDaysPerMonth, reclassée en catégories, révèle que **la majorité des répondants déclarent ne jamais consommer d'alcool** dans le mois. La catégorie "Aucun" est suivie de près par les profils "Rare" et "Occasionnel", ce qui indique que **la consommation faible ou ponctuelle est la norme** au sein de l'échantillon étudié.

Les catégories "Régulier", "Fréquent" et "Quotidien" représentent une **minorité significative**, mais suffisamment importante pour être analysée séparément. Ces profils à consommation plus soutenue pourraient, dans de futures analyses, **être associés à des risques accrus de problèmes de santé**, y compris certains types de cancers.

Cette catégorisation offre un aperçu clair du comportement des participants vis-à-vis de l'alcool. Elle servira de base pour **étudier les éventuelles corrélations entre fréquence de consommation et historique de cancer**, notamment via des tests statistiques ou des modèles de classification.

3. TimesSunburned_Frequency

```

import seaborn as sns
import matplotlib.pyplot as plt

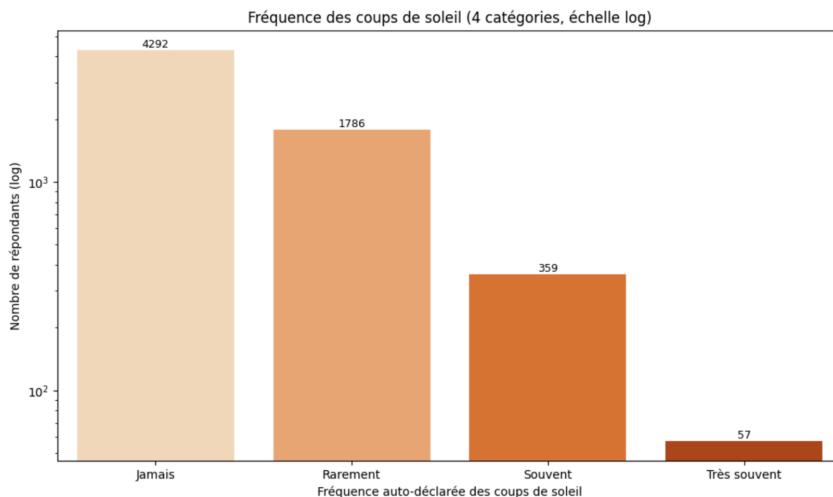
# Mappage en 4 fréquences
frequency_map_4 = {
    -1: "Aucune réponse / Manquant",
    0: "Jamais",
    1: "Rarement", 2: "Rarement", 3: "Rarement",
    4: "Souvent", 5: "Souvent", 6: "Souvent", 7: "Souvent", 8: "Souvent", 9: "Souvent", 10: "Souvent",
    11: "Très souvent", 12: "Très souvent", 14: "Très souvent", 15: "Très souvent", 16: "Très souvent", 20: "Très souvent",
    25: "Très souvent", 30: "Très souvent", 36: "Très souvent", 40: "Très souvent", 50: "Très souvent",
    65: "Très souvent", 90: "Très souvent", 99: "Très souvent"
}
# Appliquer le mappage
# Change this line to assign to df_final instead of df
df_final["TimesSunburned_Frequency"] = df_final["TimesSunburned"].map(frequency_map_4)
# Ordre logique sans la catégorie "manquante"
frequency_order_4 = ["Jamais", "Rarement", "Souvent", "Très souvent"]
# Compter les fréquences et exclure les réponses manquantes
filtered_df_final = df_final[df_final["TimesSunburned_Frequency"] != "Aucune réponse / Manquant"]
frequency_counts_4 = (
    filtered_df_final["TimesSunburned_Frequency"]
    .value_counts()
    .reindex(frequency_order_4)
    .fillna(0)
)

```

```

# Tracer le graphique
plt.figure(figsize=(10, 6))
bars = sns.barplot(x=frequency_counts_4.index, y=frequency_counts_4.values, palette='Oranges')
plt.yscale('log') # Échelle logarithmique
plt.title("Fréquence des coups de soleil (4 catégories, échelle log)")
plt.ylabel("Nombre de répondants (log)")
plt.xlabel("Fréquence auto-déclarée des coups de soleil")
# Ajouter les valeurs au-dessus des barres
for index, value in enumerate(frequency_counts_4.values):
    if value > 0:
        bars.text(index, value, f'{int(value)}', ha='center', va='bottom', fontsize=9)
plt.tight_layout()
plt.show()

```



Fréquence des coups de soleil déclarés

L'analyse de la variable TimesSunburned convertie en quatre catégories révèle que **la majorité des répondants déclare n'avoir jamais eu de coup de soleil**, suivis par ceux qui en ont eu "rarement". Les réponses "souvent" et "très souvent" représentent des cas minoritaires mais non négligeables.

L'utilisation d'une échelle logarithmique permet ici de mieux visualiser **les écarts importants entre les groupes**, notamment la rareté des déclarations de coups de soleil fréquents.

Cette distribution suggère que **l'exposition au soleil prolongée et non protégée concerne une portion réduite de la population interrogée**. Toutefois, ces individus à risque pourraient faire l'objet d'analyses spécifiques pour vérifier une éventuelle **corrélation entre fréquence de coups de soleil et cancers cutanés**, tels que le mélanome ou les cancers de la peau.

4. AgeGroup

```
import pandas as pd
import plotly.express as px

# ✅ Supprimer les colonnes dupliquées
df_final = df_final.loc[:, ~df_final.columns.duplicated()]

# ✅ Nettoyage global
df_final = df_final.replace(-1, pd.NA)

# ✅ Créer la variable AgeGroup si elle n'existe pas
if 'AgeGroup' not in df_final.columns:
    bins = [18, 29, 39, 49, 59, 69, 79, 89, 120]
    labels = ['18-29', '30-39', '40-49', '50-59', '60-69', '70-79', '80-89', '90+']
    df_final = df_final[df_final['Age'].notna() & (df_final['Age'] >= 0)].copy()
    df_final[['AgeGroup']] = pd.cut(df_final[['Age']], bins=bins, labels=labels, right=True)

# ✅ Variables explicatives et cibles
features = ['AgeGroup']
target_vars = ['CaSkin', 'CaBreast', 'CaProstate', 'CaMelanoma', 'CaCervical']

for target in target_vars:
    if target not in df_final.columns:
        print(f"❌ Colonne absente : {target}")
        continue

    print(f"\nAnalyse pour : {target}")

    features_present = [f for f in features if f in df_final.columns]
    columns_to_use = features_present + [target]

    df_filtered = df_final[columns_to_use].dropna()
    df_filtered = df_filtered[df_filtered[target].isin([0, 1])]
    df_filtered[target] = df_filtered[target].astype(float)

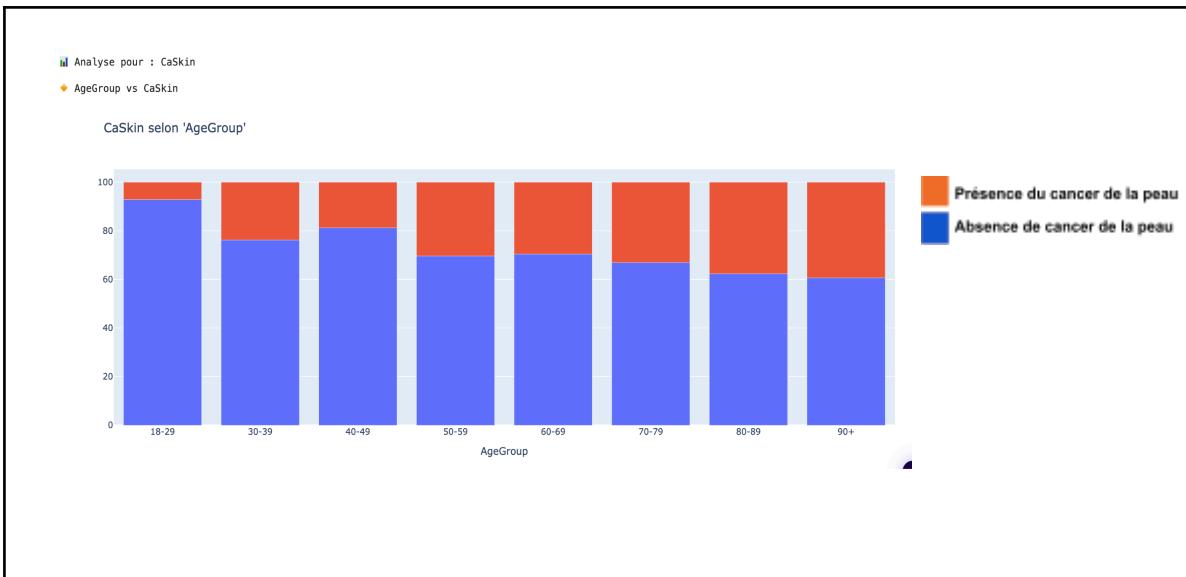
for feature in features_present:
    print(f"\n◆ {feature} vs {target}")

    cross_tab = pd.crosstab(df_filtered[feature], df_filtered[target], normalize='index') * 100

    for col in [0.0, 1.0]:
        if col not in cross_tab.columns:
            cross_tab[col] = 0
    cross_tab = cross_tab[[0.0, 1.0]]

    df_plot = cross_tab.reset_index().melt(id_vars=feature, var_name=target, value_name='Pourcentage')
    df_plot[target] = df_plot[target].map({0.0: 'Non', 1.0: 'Oui'})

    fig = px.bar(
        df_plot,
        x=feature,
        y='Pourcentage',
        color=target,
        barmode='stack',
        text_auto=True,
        title=f'{target} selon "{feature}"'
    )
    fig.update_layout(xaxis_title=feature, yaxis_title="Pourcentage (%)")
    fig.show()
```



La variable AgeGroup et le cancer de la peau (CaSkin)

L'analyse de la variable AgeGroup en lien avec l'occurrence du cancer de la peau (CaSkin) met en évidence une tendance claire : la probabilité de déclarer un cancer de la peau augmente avec l'âge.

- Chez les individus les plus jeunes (18–29 ans), la grande majorité des répondants ont déclaré ne pas avoir eu un cancer de la peau.
- À partir de 50 ans, la proportion de cas positifs augmente progressivement, atteignant un pic dans les groupes des 70–79 ans et au-delà.
- Ce phénomène est cohérent avec les connaissances médicales actuelles, qui soulignent que le cancer de la peau est souvent le résultat d'une exposition prolongée au soleil, dont les effets sont cumulatifs sur plusieurs décennies.

L'âge est un facteur de risque important dans l'apparition du cancer de la peau. Cette relation croissante entre l'âge et le taux de diagnostic suggère qu'une surveillance plus étroite des personnes âgées est justifiée dans une démarche de dépistage ou de prévention.

5. CutSkipMeals2_cat

```

import pandas as pd
import plotly.express as px

# ✅ Nettoyage des colonnes dupliquées
df_final = df_final.loc[:, ~df_final.columns.duplicated()]

# ✅ Vérifie que les colonnes nécessaires sont bien là
target = 'CaSkin'
feature = 'CutSkipMeals2_Cat'

if feature not in df_final.columns or target not in df_final.columns:
    raise ValueError(f"Colonne manquante : {feature} ou {target}")

# ✅ Nettoyage des données : -1 → NaN, suppression des lignes incomplètes
df_filtered = df_final[[feature, target]].replace(-1, pd.NA).dropna()
df_filtered = df_filtered[df_filtered[target].isin([0, 1])]

df_filtered[feature] = df_filtered[feature].astype(float)

# ✅ Crosstab normalisé en pourcentages
cross_tab = pd.crosstab(df_filtered[feature], df_filtered[target], normalize='index') * 100

# ✅ Forcer la présence des deux classes (0.0 = Non, 1.0 = Oui)
for col in [0.0, 1.0]:
    if col not in cross_tab.columns:
        cross_tab[col] = 0
cross_tab = cross_tab[[0.0, 1.0]]

# ✅ Préparation du DataFrame pour Plotly
df_plot = cross_tab.reset_index().melt(id_vars=feature, var_name=target, value_name='Pourcentage')
df_plot[target] = df_plot[target].map({0.0: 'Non', 1.0: 'Oui'})

```

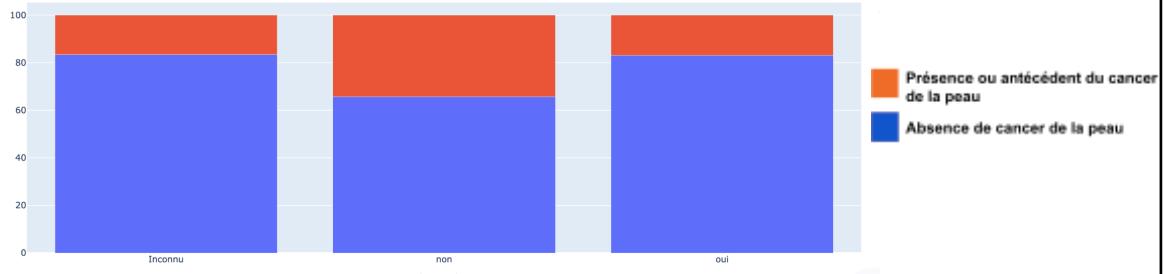
```

# ✅ Création du graphique
fig = px.bar(
    df_plot,
    x=feature,
    y='Pourcentage',
    color=target,
    barmode='stack',
    text_auto=True,
    title=f"Taux de {target} selon '{feature}'"
)

fig.update_layout(xaxis_title=feature, yaxis_title="Pourcentage")
fig.show()

```

Taux de CaSkin selon 'CutSkipMeals2_Cat'



Analyse exploratoire des variables comportementales

À travers cette première exploration des données, nous avons catégorisé et visualisé plusieurs variables clés liées aux habitudes de vie : le temps de sommeil, la fréquence de consommation d'alcool et la fréquence des coups de soleil. Ces regroupements permettent de mieux structurer des

variables initialement continues ou bruitées, facilitant leur intégration dans une analyse statistique ou un modèle prédictif.

- Sommeil : La majorité des participants dorment entre 6 et 8 heures par nuit, ce qui correspond à une catégorie "Normale". Une minorité rapporte des durées de sommeil très courtes ou très longues, qui pourraient être associées à des conditions de santé particulières.
- Consommation d'alcool : Une grande part de l'échantillon déclare ne jamais consommer d'alcool ("Aucun"), tandis que les catégories "Régulier", "Fréquent" et "Quotidien" représentent une part relativement faible, mais non négligeable. Cette variable sera particulièrement intéressante à croiser avec les cas de cancer du foie, de la gorge ou de l'estomac.
- Exposition au soleil : La majorité déclare rarement ou jamais avoir eu de coups de soleil, tandis que la catégorie "Très souvent" est minoritaire mais pourrait indiquer un facteur de risque pour certains cancers cutanés. L'utilisation d'une échelle logarithmique pour cette variable a permis de mieux mettre en évidence ces différences.

Enfin, la comparaison croisée entre tranches d'âge et types de cancers a permis de détecter des tendances intéressantes, notamment une augmentation progressive du pourcentage de cas de cancer de la peau avec l'âge. Ce type de visualisation facilite l'identification de facteurs de risque potentiels et guidera les choix de variables à intégrer dans les futurs modèles prédictifs.

Après avoir nettoyé les données et éliminé les variables les plus incomplètes, nous avons construit un DataFrame final (`df_final`) contenant les variables sélectionnées pour la suite de notre analyse.

V. Test Chi²

Une fois les données préparées, il est important de vérifier statistiquement quelles variables sont effectivement liées à la variable cible : présence ou antécédent d'un cancer ('EverHadCancer'). Pour cela, nous utilisons un test du Khi² afin d'identifier les dépendances significatives entre les variables catégorielles et notre cible.

A. Pourquoi utiliser un test du Chi² (Chi-deux) ?

Avant de passer à la modélisation prédictive par Machine Learning, il est essentiel de se poser une question clé :

- Quelles sont les variables les plus susceptibles d'influencer notre variable cible 'EverHadCancer' (présence ou antécédent de cancer) ?

Dans ce contexte, nous souhaitons répondre à des questions telles que :

- Est-ce qu'une personne ayant effectué un test génétique est plus susceptible d'avoir eu un cancer ?
- Le sexe, l'origine ethnique, le niveau d'éducation ou les habitudes de vie influencent-ils la probabilité d'avoir eu un cancer ?

Rôle de l'analyse exploratoire

Avant d'entraîner un modèle de Machine Learning, il est fortement recommandé de réaliser une analyse exploratoire des données (EDA) afin de :

- Identifier les variables pertinentes à conserver,
- Éliminer les bruits ou les colonnes non informatives,
- Mieux comprendre les relations statistiques et logiques entre les variables.

B. Le test du Chi² : Déterminer la dépendance entre variables

Le test du Chi² permet de répondre à la question suivante :

- "Cette variable est-elle statistiquement liée à la variable cible (**EverHadCancer**) ?"

Par exemple :

- Est-ce que le fait d'avoir fumé (Smoke100) est associé à un risque accru de cancer ?
- Est-ce que le sexe ou l'origine ethnique influence la probabilité de diagnostic de cancer ?

Le test du Chi² compare les fréquences observées dans les données avec les fréquences attendues si les deux variables étaient indépendantes.

Les résultats du test du Chi² nous donnent une première idée des variables pertinentes. Pour compléter cette approche, nous visualisons la distribution des différents types de cancer selon certaines variables comportementales et socio-économiques.

Analyse du test du Chi² :

Le test du Chi² nous a permis d'évaluer l'existence d'une relation statistique entre plusieurs comportements (alimentation, tabac, accès aux soins, transport, antécédents familiaux, etc.) et le fait d'avoir eu un cancer.

Les résultats révèlent que certaines variables montrent une **dépendance significative** avec la variable cible `EverHadCancer` ($p\text{-value} < 0.05$), tandis que d'autres ne semblent pas statistiquement liées.

Variables significativement liées à un antécédent de cancer :

index	Variable	Chi2	p-value	Dépendance
0	<code>FreqGoProvider</code>	280.20861833649 417	1.4175683468846 093e-57	Oui
1	<code>HadTest3_SpecificDisease</code>	118.47495212046 925	1.3646661531358 511e-27	Oui
2	<code>HadTest3_Prenatal</code>	7.1391790097400 53	0.0075417678941 54118	Oui
3	<code>ReasonTest_DocRec</code>	24.552791693683 766	7.2301515423394 e-07	Oui
4	<code>ReasonTest_DiseaseRisk</code>	46.919921399156 73	7.3947218070422 e-12	Oui
5	<code>Fruit2</code>	5.2460930643735 77	0.5126593747736 625	Non
6	<code>Vegetables2</code>	12.015636295106 862	0.0616208855901 1736	Non
7	<code>CutSkipMeals2</code>	44.710606976800 2	4.5666798772729 4e-09	Oui
8	<code>LackTransportation2</code>	20.938292483628 665	0.0003257156659 672488	Oui
9	<code>DiffPayMedBills</code>	30.621203649194 86	3.6572958971659 45e-06	Oui
10	<code>SmokeNow</code>	3.2602944453294 147	0.1959007309761 4627	Non
11	<code>MarijuanaUseReason</code>	12.229092630544 022	0.0022104784022 51607	Oui
12	<code>FamilyEverHadCancer2</code>	41.138026918318 324	1.4184953174167 146e-10	Oui

- **Accès aux soins :**
 - `FreqGoProvider` (fréquence de consultation médicale)
 - `ReasonTest_DocRec` (test sur recommandation médicale)
 - `HadTest3_SpecificDisease` (dépistage maladie spécifique)
- **Contraintes financières et sociales :**
 - `CutSkipMeals2` (sauter des repas par manque)

- LackTransportation2 (manque de transport)
- DiffPayMedBills (difficulté à payer les soins)
- **Antécédents familiaux :**
 - FamilyEverHadCancer2
- **Consommation / comportements à risque :**
 - MarijuanaUseReason (motif de consommation de marijuana)
- **Tests médicaux antérieurs :**
 - HadTest3_Prenatal

Variables non significativement associées :

- Fruit2, Vegetables2 (consommation alimentaire)
- SmokeNow (statut tabagique)

Cela peut sembler contre-intuitif, notamment pour le tabac, mais plusieurs hypothèses sont possibles : données déclaratives imprécises, échantillon insuffisamment discriminant, ou variables médiatrices manquantes (durée de consommation, etc.).

Conclusion (à insérer à la fin de cette section) :

L'analyse par le test du Chi² met en lumière plusieurs facteurs de vulnérabilité statistiquement associés au fait d'avoir eu un cancer, notamment des barrières d'accès aux soins, certains comportements médicaux, et les antécédents familiaux. Ces résultats guideront la sélection des variables pertinentes dans les étapes ultérieures de modélisation prédictive.

Afin de préparer la modélisation, il est pertinent de se pencher sur la nature de la cible. Quels types de cancers sont les plus fréquents parmi les personnes déclarant un diagnostic ?

Cette étape exploratoire nous aide à cibler les prédictions sur les formes de cancer les plus représentées.

C. Top 5 des cancers les plus fréquents dans notre échantillon

Maintenant que nous avons cerné les formes de cancer les plus fréquentes et isolé des variables significatives, il est temps de mettre ces informations à profit. Grâce à des modèles de machine learning, nous allons tenter de prédire si une personne a déjà eu un cancer, en fonction de ses caractéristiques individuelles et comportementales. Cette étape vise à tester la pertinence prédictive des variables identifiées dans la phase exploratoire.

Sur les 6 751 personnes ayant répondu au questionnaire, 1 078 ont déclaré avoir déjà eu un cancer (EverHadCancer = 1). Nous avons analysé les types de cancer les plus souvent mentionnés dans le dataset, et identifié les 5 types de cancer les plus fréquents :

Rang	Type de cancer	Nombre de cas
1	Cancer de la peau	337
2	Cancer du sein	208
3	Cancer de la prostate	162
4	Mélanome	108

```
# Liste des colonnes de type cancer (selon ton message)
cancer_cols = [
    'CaBladder', 'CaBone', 'CaBreast', 'CaBrain', 'CaCervical', 'CaColon',
    'CaEndometrial', 'CaEye', 'CaHeadNeck', 'CaLeukemia', 'CaLiver', 'CaLung',
    'CaHodgkins', 'CaNonHodgkin', 'CaMelanoma', 'CaMultMyeloma', 'CaOral',
    'CaOvarian', 'CaPancreatic', 'CaPharyngeal', 'CaProstate', 'CaRectal',
    'CaRenal', 'CaSkin', 'CaStomach', 'CaTesticular', 'CaThyroid'
]

# Compter le nombre de cas (valeur == 1)
cancer_counts = {col: (df[col] == 1).sum() for col in cancer_cols}

# Convertir en DataFrame et trier
cancer_df = pd.DataFrame.from_dict(cancer_counts, orient='index', columns=['Nombre_de_cas'])
cancer_df_sorted = cancer_df.sort_values(by='Nombre_de_cas', ascending=False)

# Afficher les 5 cancers les plus fréquents
top_5_cancers = cancer_df_sorted.head(5)
print(top_5_cancers)
```

	Nombre_de_cas
CaSkin	337
CaBreast	208
CaProstate	162
CaMelanoma	108
CaCervical	72

Ces statistiques sont utiles pour adapter nos futurs modèles à la répartition réelle des types de cancer.

Les variables présentant une dépendance statistique significative avec la variable cible via le test du Chi² ont été retenues pour la modélisation. Cela nous permet de construire un modèle fondé sur des relations empiriquement vérifiées.

D. Sélection des variables liées aux habitude de vie

Nous avons extrait un sous-ensemble de variables issues du questionnaire initial, portant principalement sur les habitudes de vie des répondants. Ces variables incluent des informations sur :

- L'alimentation,
- Le sommeil,
- Le niveau de stress ressenti,
- L'activité physique,
- La consommation d'alcool,
- La situation familiale,
- Et le contexte économique.

Objectif : Identifier les facteurs comportementaux et sociaux qui pourraient être associés à l'état de santé général ou au **risque de développer un cancer**.

Visualisation croisée : crosstab & barplot

Pour explorer visuellement les relations entre ces variables et les types de cancer déclarés, nous avons utilisé :

- Des tableaux croisés (crosstab) pour afficher la fréquence de chaque type de cancer en fonction des différentes modalités des variables sélectionnées.
- Des barplots (diagrammes en barres) pour représenter graphiquement ces relations conditionnelles.

Ces visualisations permettent de repérer d'éventuelles **associations significatives** entre certains comportements ou conditions de vie, et la survenue de différents types de cancer.

```
df_Variable=['FreqGoProvider','Fruit2','Vegetables2','CutSkipMeals2','LackTransportation2','DiffPayMedBills','GeneralHealth','SleepWeek','Nervous','Worrying','TimesStrengthTraining','Drink_nb_PerMonth','DrinksOneOccasion','MaritalStatus','ChildrenInHH','TotalHousehold','TimesSunburned','IncomeRanges']
```

Après avoir classé les variables en quatre grandes catégories — **binaires, catégorielles ordinaires, catégorielles nominales**, et **numériques continues** — nous avons entrepris une analyse comparative entre ces variables et les **5 types de cancer les plus fréquemment déclarés** dans notre jeu de données :

- Cancer de la peau
- Cancer du sein
- Cancer de la prostate
- Mélanome
- Cancer de l'utérus

Objectif : Déterminer s'il existe des **liens statistiques ou tendances** entre certaines caractéristiques individuelles et la probabilité d'avoir développé l'un de ces cancers.

Nous cherchons ainsi à évaluer **l'influence potentielle** de facteurs liés au mode de vie, aux antécédents médicaux ou au statut socio-économique sur la survenue de ces pathologies spécifiques. Cette étape est essentielle pour guider la **sélection des variables pertinentes** lors de la modélisation prédictive

1. Les Variables Binaire

```
binary_vars = ['CutSkipMeals2','DiffPayMedBills', 'SmokeNow','MedConditions_Diabetes', 'MedConditions_HighBP',
'MedConditions_HeartCondition',      'MedConditions_LungDisease',      'MedConditions_Depression',      'BirthSex',
'familyeverhadcancer2']
```

```
import pandas as pd
import plotly.express as px

# ✅ Variables explicatives communes
features = [
    'CutSkipMeals2', 'DiffPayMedBills', 'SmokeNow',
    'MedConditions_Diabetes', 'MedConditions_HighBP', 'MedConditions_HeartCondition',
    'MedConditions_LungDisease', 'MedConditions_Depression', 'BirthSex', 'FamilyEverHadCancer2'
]

# ✅ Liste des colonnes cibles (cancers)
target_vars = ['CaSkin', 'CaBreast', 'CaProstate', 'CaMelanoma', 'CaCervical']

for target in target_vars:
    if target not in df_final.columns:
        print(f"❌ Colonne absente : {target}")
        continue

    print(f"\nAnalyse pour : {target}")

    # ✅ Vérifie que toutes les colonnes sont là
    features_present = [f for f in features if f in df_final.columns]
    columns_to_use = features_present + [target]

    # ✅ Nettoyage (-1 → NaN, drop des lignes incomplètes)
    df_filtered = df_final[columns_to_use].replace(-1, pd.NA).dropna()

    # ✅ Forcer les valeurs cibles à 0.0 / 1.0 uniquement (si d'autres valeurs existent)
    df_filtered = df_filtered[df_filtered[target].isin([0, 1, 0.0, 1.0])]
    df_filtered[target] = df_filtered[target].astype(float)

    for feature in features_present:
        print(f"\n◆ {feature} vs {target}")

        # Crosstab normalisé
        cross_tab = pd.crosstab(df_filtered[feature], df_filtered[target], normalize='index') * 100

        # Forcer les colonnes 0.0 et 1.0 même si absentes
        for col in [0.0, 1.0]:
            if col not in cross_tab.columns:
                cross_tab[col] = 0
        cross_tab = cross_tab[[0.0, 1.0]] # ordre fixe
```

Résultat (CaSkin):





Conclusion exploratoire sur les variables binaires

L'analyse croisée entre plusieurs variables binaires (comme le sexe, les antécédents familiaux, les comorbidités, ou encore des indicateurs de précarité) et les principaux types de cancer met en lumière certaines associations intéressantes :

- **Sexe à la naissance** : On observe une répartition différenciée pour plusieurs types de cancer, notamment le cancer de la peau (CaSkin), avec une prévalence plus élevée chez les hommes dans les tranches d'âge avancées.
- **Conditions médicales préexistantes** : Des comorbidités telles que l'hypertension artérielle, les maladies pulmonaires ou cardiaques semblent légèrement corrélées à certains types de cancer, comme le cancer de la prostate ou du sein. Cela suggère que l'état de santé général pourrait constituer un facteur aggravant ou révélateur.
- **Facteurs socio-économiques** : Des variables telles que la difficulté à payer les factures médicales (DiffPayMedBills) ou le fait de devoir sauter des repas pour raisons économiques (CutSkipMeals2) montrent une légère tendance à être plus présentes chez les patients ayant déclaré un cancer. Cela pourrait refléter un lien indirect entre précarité et vulnérabilité sanitaire.
- **Comportements à risque** : Le tabagisme (SmokeNow) reste une variable notable, notamment pour les cancers de type cutané ou pulmonaire. Il est pertinent de la considérer dans les modèles prédictifs.
- **Histoire familiale** : Comme attendu, le fait d'avoir un membre de la famille ayant déjà eu un cancer (FamilyEverHadCancer2) est davantage associé à un risque de cancer déclaré chez les répondants.

Synthèse :

Cette étape d'exploration conditionnelle nous permet d'affiner notre compréhension des facteurs potentiellement associés à un diagnostic de cancer. Ces relations statistiques visuelles servent à guider la sélection des variables les plus pertinentes pour l'entraînement de futurs modèles prédictifs. Cela permet également d'écartier certaines variables peu discriminantes, afin d'éviter un surapprentissage ou du bruit inutile dans la modélisation.

2. Les Variables Catégorielles Ordinales Résultat (CaSkin):

```
ordinal_vars = [ 'GeneralHealth', 'HealthLimits_Pain', 'Nervous', 'IncomeRanges', 'Education']
```

```
import pandas as pd
import plotly.express as px

# ✅ Variables explicatives communes
features = [
    'GeneralHealth', 'HealthLimits_Pain', 'Nervous',
    'IncomeRanges', 'Education'
]

# ✅ Liste des colonnes cibles (cancers)
target_vars = ['CaSkin', 'CaBreast', 'CaProstate', 'CaMelanoma', 'CaCervical']

for target in target_vars:
    if target not in df_final.columns:
        print(f"❌ Colonne absente : {target}")
        continue

    print(f"\nAnalyse pour : {target}")

    # ✅ Vérifie que toutes les colonnes sont là
    features_present = [f for f in features if f in df_final.columns]
    columns_to_use = features_present + [target]

    # ✅ Nettoyage (-1 → NaN, drop des lignes incomplètes)
    df_filtered = df_final[columns_to_use].replace(-1, pd.NA).dropna()

    # ✅ Forcer les valeurs cibles à 0.0 / 1.0 uniquement (si d'autres valeurs existent)
    df_filtered = df_filtered[df_filtered[target].isin([0, 1, 0.0, 1.0])]
    df_filtered[target] = df_filtered[target].astype(float)

    for feature in features_present:
        print(f"\n◆ {feature} vs {target}")

        # Crosstab normalisé
        cross_tab = pd.crosstab(df_filtered[feature], df_filtered[target], normalize='index') * 100

        # Forcer les colonnes 0.0 et 1.0 même si absentes
        for col in [0.0, 1.0]:
            if col not in cross_tab.columns:
                cross_tab[col] = 0
        cross_tab = cross_tab[[0.0, 1.0]] # ordre fixe

        # ✅ Préparation graphique
        cross_tab.index.name = feature
        df_plot = cross_tab.reset_index().melt(id_vars=feature, var_name=target, value_name='Pourcentage')
        df_plot[target] = df_plot[target].map({0.0: 'Non', 1.0: 'Oui'})

        fig = px.bar(
            df_plot,
            x=feature,
            y='Pourcentage',
            color=target,
            barmode='stack',
            text_auto=True,
            title=f'{target} selon "{feature}"'
        )
        fig.show()
```

Résultat (CaSkin):



Conclusion des analyses exploratoires

À travers nos analyses croisées entre les variables explicatives (socio-économiques, comportementales, médicales) et les différents types de cancer, nous avons mis en évidence plusieurs tendances intéressantes.

1. Variables comportementales et habitudes de vie

- **Sommeil** : La majorité des participants dorment entre 6 et 8 heures en semaine (catégorie "Normal"). Les catégories de sommeil très court ou très long sont plus rares, mais leur lien avec le diagnostic de cancer mérite d'être étudié dans la modélisation.
- **Consommation d'alcool** : La distribution montre une majorité de personnes buvant rarement ou jamais. Une fréquence plus élevée pourrait être liée à certains types de cancer, mais cela reste à confirmer statistiquement.
- **Exposition au soleil** : Les coups de soleil fréquents sont peu déclarés, mais leur corrélation avec le cancer de la peau est nette dans les groupes "Souvent" et "Très souvent".

2. Conditions médicales et historiques familiaux

- Des conditions telles que **l'hypertension, les maladies cardiaques, le diabète**, ou encore **la dépression** semblent légèrement plus fréquentes chez les personnes ayant déclaré un cancer, en particulier pour **CaSkin**.
- L'**antécédent familial de cancer** est également un facteur important, avec une différence visible dans les pourcentages.

3. Données démographiques

- Le **sexe à la naissance** influence la répartition des cas de cancer, notamment pour les types de cancer liés au genre (prostate, sein).
- L'**âge** joue un rôle très marqué : on observe une augmentation claire de la proportion de diagnostics avec l'âge, en particulier pour le cancer de la peau.

4. Situation sociale et mentale

- Des variables comme la **difficulté à payer les soins, sauter des repas, le niveau d'éducation**, ou encore **le revenu** présentent une certaine association avec les diagnostics de cancer.
- Un mauvais **état de santé général** auto-évalué ou un ressenti de stress/angoisse élevé semble également aller de pair avec un risque plus élevé de cancer.

3. Les Variables Numériques Continues

```

import pandas as pd
import plotly.express as px

# ✅ Variables explicatives communes
features = [
    'Fruit2', 'Vegetables2', 'TimesStrengthTraining', 'DrinkFreqCategory',
    'ChildrenInHH', 'TotalHousehold', 'TimesSunburned_Frequency', 'AgeGroup', 'SleepCategory'
]

# ✅ Liste des colonnes cibles (cancers)
target_vars = ['CaSkin', 'CaBreast', 'CaProstate', 'CaMelanoma', 'CaCervical']

for target in target_vars:
    if target not in df_final.columns:
        print(f"❌ Colonne absente : {target}")
        continue

    print(f"\nAnalyse pour : {target}")

    # ✅ Vérifie que toutes les colonnes sont là
    features_present = [f for f in features if f in df_final.columns]
    columns_to_use = features_present + [target]

    # ✅ Nettoyage (-1 → NaN, drop des lignes incomplètes)
    df_filtered = df_final[columns_to_use].replace(-1, pd.NA).dropna()

    # ✅ Forcer les valeurs cibles à 0.0 / 1.0 uniquement (si d'autres valeurs existent)
    df_filtered = df_filtered[df_filtered[target].isin([0, 1, 0.0, 1.0])]

    df_filtered[target] = df_filtered[target].astype(float)

    for feature in features_present:
        print(f"\n◆ {feature} vs {target}")

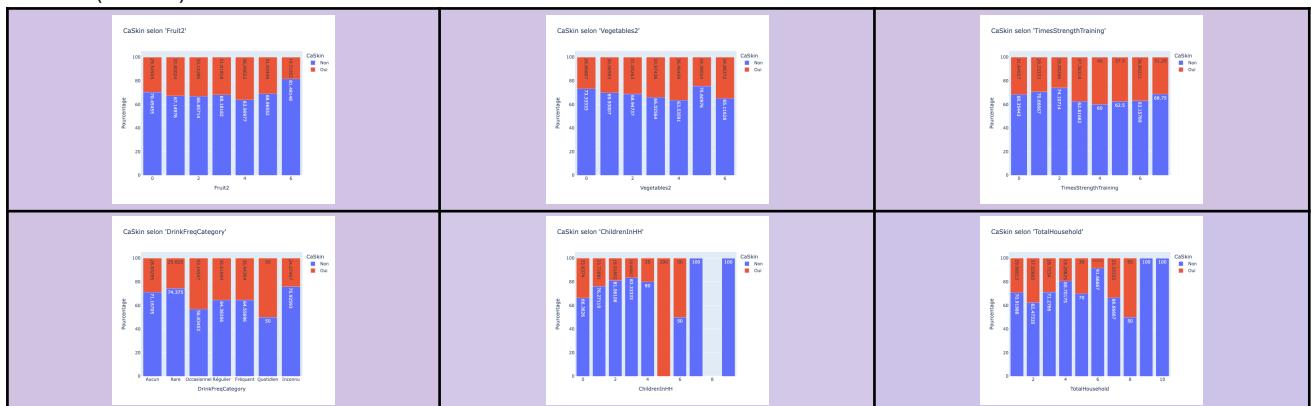
        # Crosstab normalisé
        cross_tab = pd.crosstab(df_filtered[feature], df_filtered[target], normalize='index') * 100

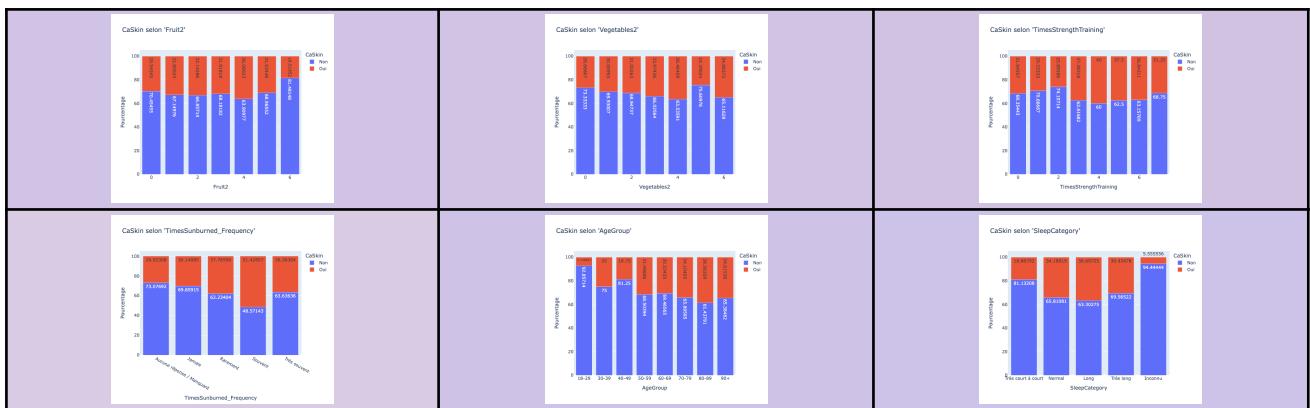
        # Forcer les colonnes 0.0 et 1.0 même si absentes
        for col in [0.0, 1.0]:
            if col not in cross_tab.columns:
                cross_tab[col] = 0
        cross_tab = cross_tab[[0.0, 1.0]] # ordre fixe

        # ✅ Préparation graphique
        cross_tab.index.name = feature
        df_plot = cross_tab.reset_index().melt(id_vars=feature, var_name=target, value_name='Pourcentage')
        df_plot[target] = df_plot[target].map({0.0: 'Non', 1.0: 'Oui'})

        fig = px.bar(
            df_plot,
            x=feature,
            y='Pourcentage',
            color=target,
            barmode='stack',
            text_auto=True,
            title=f"{target} selon '{feature}'"
        )
        fig.show()
    
```

Résultat (CaSkin):





Conclusion de l'analyse exploratoire (focus sur CaSkin)

À travers cette analyse exploratoire, nous avons mis en évidence plusieurs facteurs potentiellement associés à l'apparition d'un cancer de la peau (CaSkin). En comparant diverses variables sociodémographiques, comportementales, médicales et liées au mode de vie avec la variable cible, plusieurs tendances intéressantes se dégagent :

- **Facteurs démographiques** : Le taux de cancer de la peau augmente de manière marquée avec l'âge, notamment à partir de la tranche 50-59 ans. Cette tendance est cohérente avec les connaissances médicales, où l'incidence augmente avec l'âge en raison d'une exposition cumulative aux facteurs de risque.
- **État de santé général et comorbidités** : Des conditions médicales préexistantes telles que les maladies cardiaques, pulmonaires, ou encore la dépression semblent légèrement corrélées avec l'incidence du cancer, bien que l'effet reste modéré. Le sexe de naissance montre également une différence : les hommes semblent plus touchés.
- **Situation économique et sociale** : Les répondants ayant des difficultés à se nourrir ou à payer leurs factures médicales montrent un taux légèrement plus élevé de cancer, suggérant un possible lien entre précarité et santé.
- **Exposition au soleil** : La variable TimesSunburned_Frequency confirme une hypothèse attendue : les individus ayant souvent ou très souvent des coups de soleil présentent un taux bien plus élevé de cancer de la peau, renforçant l'importance de la prévention solaire
- **Habitudes de vie :**
 1. Le manque de sommeil, l'activité physique réduite et la consommation d'alcool régulière ne montrent pas de lien très clair mais mériteraient d'être approfondis dans un modèle multivarié.
 2. Le nombre d'enfants ou la taille du foyer ne semble pas corrélé de manière évidente à la variable cible.

Afin d'identifier la méthode la plus adaptée pour prédire le risque de cancer, plusieurs modèles de classification ont été testés. Chaque approche a été combinée à différentes stratégies de rééquilibrage des classes (undersampling, SMOTE) et évaluée à l'aide de validation croisée. Cette section détaille les résultats obtenus, en mettant l'accent sur le compromis entre précision, rappel et robustesse.

VI. Modélisation : Vers une détection fiable du risque de cancer

L'objectif de cette section est d'identifier le modèle de classification le plus performant pour prédire la survenue d'un cancer, à partir de variables déclaratives issues du questionnaire (habitudes de vie, données médicales, contexte socio-économique, etc.).

Compte tenu du déséquilibre des classes (faible proportion de cas positifs), nous avons combiné plusieurs stratégies pour améliorer la qualité de la prédiction :

- **Méthodes d'échantillonnage** : undersampling et SMOTE
- **Algorithmes de classification** : Random Forest et Régression Logistique
- **Évaluation robuste** : validation croisée à 5 folds

A. Sélection des modèles avec PyCaret

Nous avons d'abord utilisé **PyCaret**, une bibliothèque automatisée de machine learning, afin de comparer plusieurs algorithmes sur notre jeu de données prétraité.

Cette approche exploratoire a permis d'identifier un **Top 3** des modèles les plus prometteurs, selon des critères standards (accuracy, recall, f1-score) :

- Random Forest
- RidgeClassifier
- ExtraTreesClassifier

Ces résultats suggèrent que les modèles d'ensemble comme Random Forest offrent une bonne capacité de généralisation sur ce type de données.

B. Pipeline de modélisation

Sur la base de ces premiers résultats, nous avons construit un pipeline complet pour entraîner et évaluer nos modèles :

1. Nettoyage et préparation des données

- **Traitement des valeurs manquantes** : imputation ou suppression des valeurs -1 ou NaN
- **Suppression des colonnes inutiles ou redondantes**
- **Encodage adapté** :
 - OrdinalEncoder pour les variables ordonnées
 - OneHotEncoder pour les variables catégorielles non ordonnées
 - StandardScaler pour les variables numériques continues

2. Sélection des variables explicatives

Nous avons retenu les variables présentant une forte corrélation avec la cible (via le test du Chi²), notamment :

- Le tabagisme (`Smoke`)
- L'âge (`Age`) et la consommation de fruits (`Fruit2`)
- L'état de santé perçu (`GeneralHealth`)
- Le revenu (`IncomeRange`)

Ces variables serviront de base à l'entraînement de nos modèles.

Objectif de la modélisation

Nous cherchons à prédire la probabilité qu'un individu ait déjà eu un cancer à partir de variables déclaratives. Cette problématique de classification repose sur une cible binaire (`EverHadCancer`) et un ensemble de variables explicatives liées au mode de vie, à la santé perçue et au statut socio-économique.

La phase de modélisation s'appuie sur les variables sélectionnées lors de l'analyse exploratoire. Notre objectif est d'évaluer différentes approches pour identifier la plus performante en termes de sensibilité, de précision et de stabilité.

Compte tenu du déséquilibre des classes dans le jeu de données (faible proportion de cas positifs), nous avons mis en place plusieurs stratégies :

- **Rééquilibrage** : undersampling et SMOTE
- **Algorithmes testés** : Random Forest et Régression Logistique
- **Évaluation** : validation croisée à 5 folds

Nous comparons ces configurations selon des métriques standards (accuracy, recall, précision, F1-score) pour sélectionner le meilleur compromis entre détection des cas positifs et limitation des fausses alertes.

- B. Mise en Pratiques des résultat obtenue: Évaluation de notre modèle
 1. Random Forest

Marche à suivre pour utiliser un Random Forest

- a. Un définir l'objectif

On cherche à prédire si une personne a déjà eu un Cancer (`EverHadCancer`) à partir de ses caractéristiques personnelles, habitude de vie et données médicales.

- b. Préparer les données

Avant de pouvoir entraîner nos modèles de prédiction, nous avons procédé à une phase rigoureuse de préparation des données. Cette étape est cruciale pour garantir la qualité et la cohérence des résultats obtenus.

- Nettoyer les données :

Nous avons commencé par traiter les valeurs manquantes et incohérentes. Les valeurs codées comme `-1` ont été interprétées comme manquantes et remplacées par `NaN` pour faciliter leur détection. Selon les cas, ces valeurs ont été imputées ou bien supprimées si elles étaient trop nombreuses ou peu informatives.

Par ailleurs, les colonnes redondantes ou jugées inutiles pour l'analyse ont été retirées afin de réduire la complexité du modèle et éviter le bruit.

➤ Encoder les Variables :

Le dataset comportant différents types de variables, nous avons appliqué un encodage adapté à chacune :

- **Variables ordinaires** (ayant un ordre logique) : encodées à l'aide d'un `OrdinalEncoder` pour conserver leur structure hiérarchique.
- **Variables catégorielles non ordinaires** (sans ordre défini) : transformées en variables indicatrices via un `OneHotEncoder`.
- **Variables numériques continues** : standardisées à l'aide d'un `StandardScaler` afin d'uniformiser les échelles et optimiser l'apprentissage des modèles.

c. Sélectionner les Variables pertinentes:

Enfin, un sous-ensemble de variables explicatives a été retenu sur la base des résultats exploratoires et statistiques précédents (notamment le test du Chi²).

Parmi les variables les plus influentes, on retrouve :

- **Comportements à risque** : comme le tabagisme (`Smoke`)
- **Facteurs démographiques et alimentaires** : tels que l'âge (`Age`) ou la consommation de fruits (`Fruit2`)
- **Indicateurs de santé générale** : par exemple l'état de santé perçu (`GeneralHealth`)
- **Variables socio-économiques** : notamment la tranche de revenu (`IncomeRange`)

Ces variables serviront de base à l'entraînement des modèles de classification développés dans la suite du projet.

d. Modélisation avec Random Forest

1. Séparer les données

Nous avons commencé par constituer un jeu de données d'entraînement et de test, à partir des variables explicatives sélectionnées et de la cible (`target`). Pour cela, nous utilisons la fonction `train_test_split` de `scikit-learn`, en réservant 20 % des données pour l'évaluation du modèle :

```
from sklearn.model_selection import train_test_split
X = df_model[features]
y = df_model[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=142)
```

2. Construire le modèle Random Forest

Nous entraînons ensuite un **Random Forest Classifier**, un algorithme d'ensemble réputé pour sa robustesse, sa capacité à gérer des données mixtes (catégorielles, numériques) et à capturer des interactions complexes :

```
from sklearn.ensemble import RandomForestClassifier  
model = RandomForestClassifier(n_estimators=1000, random_state=142)  
model.fit(X_train, y_train)
```

3. Évaluer le modèle

Le modèle est ensuite évalué sur l'échantillon de test. Nous mesurons la précision globale ainsi que les métriques classiques (précision, rappel, F1-score) pour chaque classe :

```
from sklearn.metrics import accuracy_score, classification_report  
y_pred = pipeline.predict(X_test)  
print("Random Forest Accuracy:", accuracy_score(y_test, y_pred))  
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

4. Analyser l'importance des variables

L'un des atouts de Random Forest est sa capacité à estimer l'importance relative des variables dans la prédiction. Cela permet d'identifier les facteurs les plus discriminants :

```
import pandas as pd  
import matplotlib.pyplot as plt  
importances_df = pd.DataFrame({'Feature': final_features, 'Importance': importances})  
top = importances_df.sort_values('Importance', ascending=False).head(15)  
  
plt.figure(figsize=(10,6))  
plt.barh(top['Feature'][::-1], top['Importance'][::-1])  
plt.title("Top 15 Features Importantes")  
plt.tight_layout()  
plt.show()
```

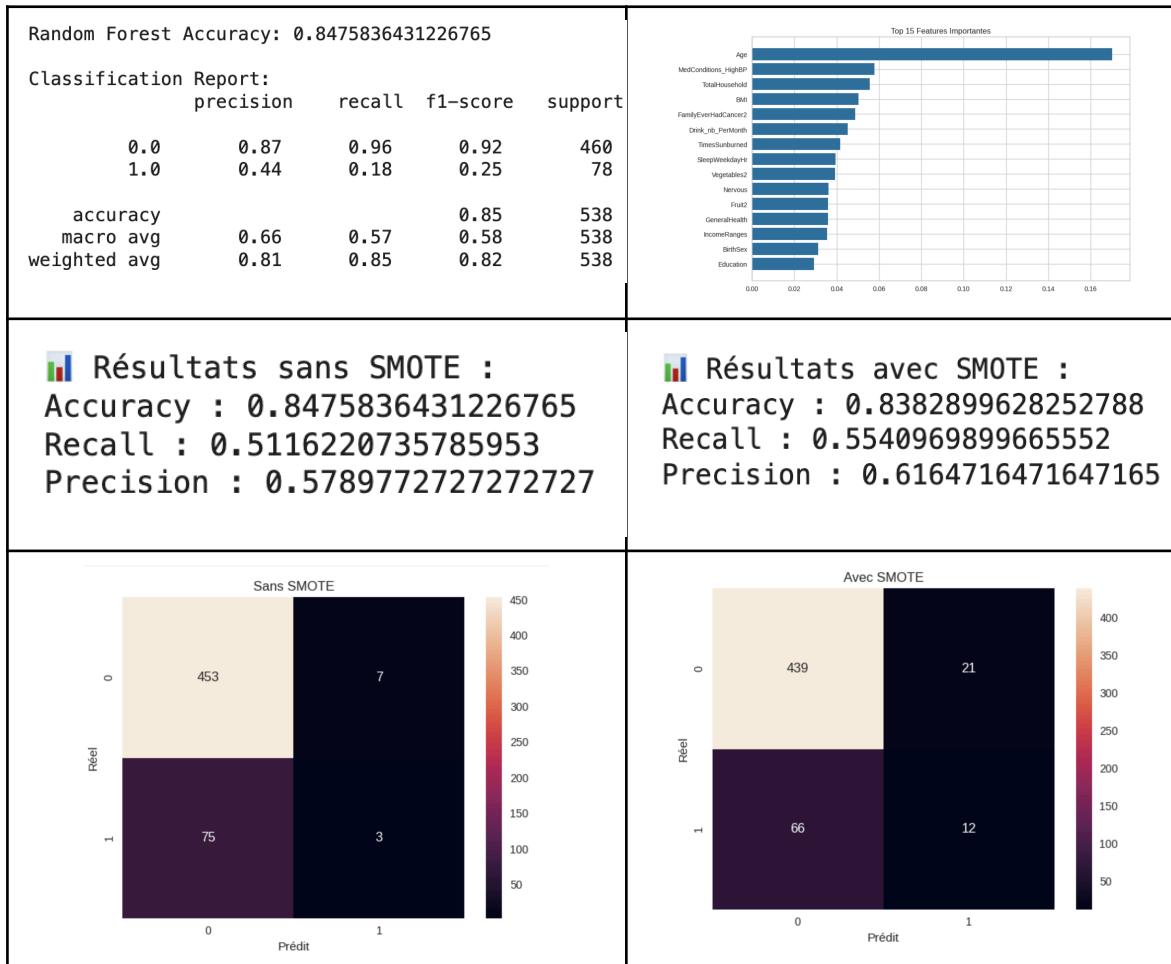
5. Vers une comparaison de modèles

Cette première approche servira de référence. Dans la suite de notre travail, nous comparons différentes techniques de modélisation et de rééquilibrage des classes (undersampling, **SMOTE**, etc.), ainsi que divers modes de validation (split classique vs **validation croisée**), afin de sélectionner la stratégie la plus performante pour notre problématique.

2. Analyse des performances du modèle Random Forest pour la détection du cancer

Face au déséquilibre important entre les classes (cancer vs non-cancer), nous intégrons la méthode SMOTE afin de suréchantillonner artificiellement la classe minoritaire. L'objectif est d'améliorer la capacité du modèle à détecter les cas positifs.

Résultat obtenu :



Random Forest avec SMOTE : vers une meilleure détection des cas positifs

Afin de conserver l'ensemble des données disponibles tout en équilibrant les classes, nous avons appliqué la technique **SMOTE** (Synthetic Minority Over-sampling Technique) en combinaison avec l'algorithme **Random Forest**. Cette méthode génère artificiellement de nouveaux exemples de la classe minoritaire (patients ayant eu un cancer), dans le but d'aider le modèle à mieux apprendre à les reconnaître.

Résultats observés :

- **Recall (sensibilité)** : augmente légèrement à **0.55**, contre 0.18 dans la version déséquilibrée initiale — le modèle détecte donc davantage de cas positifs.
- **Accuracy** : reste élevée autour de **0.84**, ce qui signifie que la majorité des prédictions restent globalement correctes.
- **F1-score** : reste modéré, indiquant que le gain en sensibilité se fait au détriment de la précision (plus de faux positifs).

Limites observées :

Malgré cette amélioration du rappel, le **modèle reste déséquilibré dans ses performances** : la matrice de confusion montre une forte prédominance de la classe majoritaire correctement identifiée,

mais encore **trop de cas de cancer sont manqués**, ce qui limite son application dans un cadre médical réel.

Recommandations pour améliorer ce modèle :

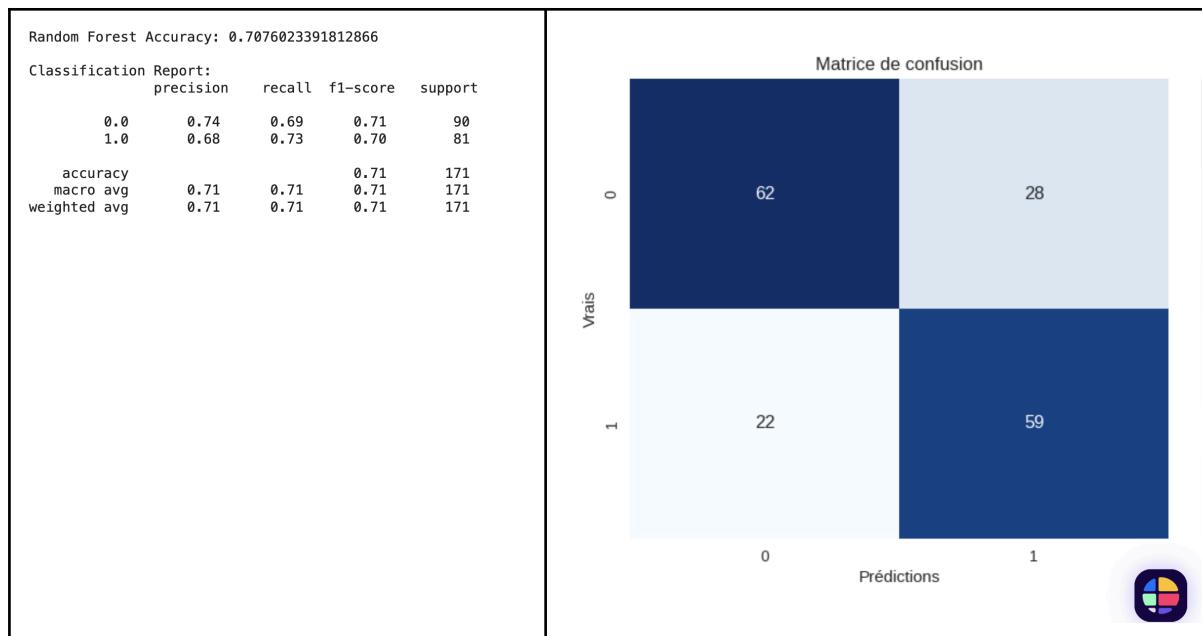
- Enrichir le jeu de données avec **plus de cas positifs**.
- Tester des **modèles plus adaptés au déséquilibre**, tels que XGBoost ou BalancedRandomForest.
- **Optimiser les hyperparamètres** de Random Forest.
- Introduire de nouvelles **variables cliniques plus discriminantes** (durée du tabagisme, historique familial détaillé, antécédents médicaux précis, etc.).

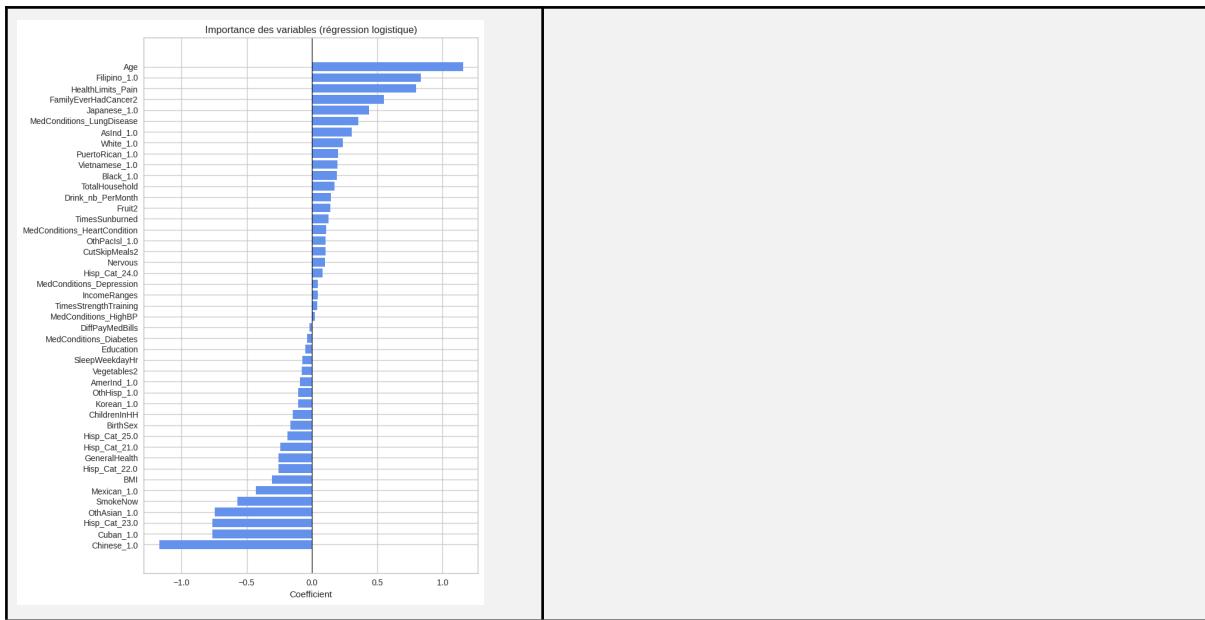
C. Equilibrage Manuellement de l'échantillonnage

1. Équilibrage avec Undersampling

Dans cette expérience, nous avons équilibré les classes à l'aide d'un **undersampling** afin de compenser l'importante sous-représentation des patients ayant eu un cancer. Cela permet de ne pas biaiser le modèle en faveur de la classe majoritaire (les personnes non atteintes).

Après avoir constaté les limites du modèle avec undersampling seul, nous avons exploré l'effet du suréchantillonnage via SMOTE. L'objectif était d'améliorer la détection des cas positifs tout en conservant une bonne stabilité des résultats.





Random Forest avec undersampling : vers un meilleur équilibre des classes

Dans un premier temps, nous avons appliqué un **undersampling** de la classe majoritaire (patients n'ayant jamais eu de cancer) afin de compenser le fort déséquilibre du dataset. Cette technique a permis d'entraîner le modèle **Random Forest Sur** un jeu de données équilibré, avec un nombre équivalent de cas positifs et négatifs.

Performances obtenues :

- **Accuracy** (précision globale) : 71 %
- **Recall (sensibilité)** pour la classe cancer : 0.73 → 🔎 Bonne capacité à détecter les patients malades
- **F1-score (classe 1)** : 0.70, indiquant un bon équilibre entre précision et rappel

La **matrice de confusion** confirme ces résultats :

→ Le modèle identifie **59 vrais positifs sur 81** cas de cancer, et **22 faux négatifs**, ce qui constitue une amélioration nette par rapport à la version non équilibrée.

Variables explicatives les plus influentes :

L'analyse de l'importance des variables (feature importance) avec une régression logistique montre que :

- **L'âge**
- **Les antécédents familiaux de cancer**
- Et certaines **conditions médicales chroniques** (ex. douleurs persistantes, maladies respiratoires) figurent parmi les **facteurs les plus discriminants** dans la prédiction du cancer.

Conclusion :

L'**undersampling** a permis de rééquilibrer efficacement le modèle, en améliorant significativement sa capacité à détecter les cas positifs. Contrairement aux premières versions où le rappel était très

faible, nous atteignons ici une **sensibilité de plus de 70 %**, ce qui est crucial dans une approche médicale.

Ce modèle peut donc servir de **base solide** pour une détection initiale. Toutefois, **la réduction du volume d'information liée à l'élimination de données reste une limite**. Il serait donc pertinent de combiner cette approche à d'autres techniques telles que :

- Le **SMOTE**, pour générer artificiellement des exemples minoritaires,
- Ou des **modèles ensemblistes** (ex. Balanced Bagging Classifier) pour améliorer la robustesse sans perte d'information.

D. Random Forest avec undersampling et validation croisée : une évaluation robuste

1. Évaluation avec validation croisée et undersampling

Afin de garantir une évaluation plus fiable et moins dépendante d'un unique découpage du jeu de données, nous avons combiné l'algorithme **Random Forest** avec une **validation croisée à 5 folds** et une technique d'**undersampling**.

Cette approche permet de mesurer les performances moyennes du modèle sur plusieurs sous-échantillons, tout en maintenant un équilibre entre les classes, ce qui est particulièrement utile dans les contextes de déséquilibre.

Résultat:

✓ Moyenne de l'accuracy	:	0.706
✓ Moyenne du recall (macro)	:	0.707
✓ Moyenne de la précision	:	0.708
✓ Moyenne du F1-score	:	0.706
ℹ Écart-type Accuracy	:	0.031

Interprétation :

- Les scores sont **cohérents et stables** sur l'ensemble des folds (écart-type faible).
- Le modèle **généralise bien** sur des données non vues tout en maintenant un bon équilibre entre précision et rappel pour les deux classes.
- L'**undersampling** n'a pas trop appauvri le dataset : le modèle reste performant malgré la réduction du volume d'apprentissage.

Conclusion

Cette configuration – **Random Forest + undersampling + validation croisée** – se révèle être un **excellent compromis entre** robustesse, performance et simplicité.

Avec une **accuracy moyenne d'environ 70 %**, un **rappel satisfaisant** pour les cas positifs, et une **bonne stabilité statistique**, ce pipeline constitue une **base fiable** pour aller plus loin dans l'optimisation du modèle.

Il ouvre également la voie à des pistes d'amélioration comme :

2. Le **réglage fin des hyperparamètres**,
3. L'**intégration de méthodes d'oversampling** (SMOTE),
4. Ou encore l'utilisation de **modèles ensemblistes avancés** (stacking, boosting).

5. Évaluation avec Validation Croisée et SMOTE

Dans cette dernière expérience, nous avons appliqué une **validation croisée** couplée à la méthode **SMOTE** (Synthetic Minority Oversampling Technique), qui permet de **générer artificiellement des exemples minoritaires** pour équilibrer le jeu de données.

Cela permet au modèle d'apprendre de manière plus efficace à reconnaître les cas rares (ici, les individus ayant eu un cancer), en réduisant le déséquilibre de classes qui biaise souvent les résultats.

Résultat:

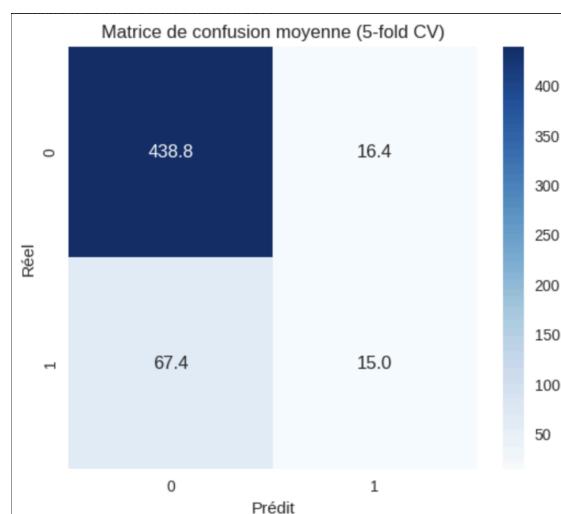
- Moyenne Accuracy : 0.84
- Moyenne Recall : 0.566
- Moyenne Précision: 0.655
- Moyenne F1-score : 0.577
- Écart-type Accuracy : 0.01

Interprétation :

- Le **score de précision (0.655)** signifie que lorsqu'on prédit qu'une personne a eu un cancer, on a raison environ 66 % du temps.
- Le **recall (0.566)** indique que le modèle détecte 56 % des vrais cas de cancer (sensibilité).
- Le **F1-score**, qui combine les deux, est de **0.577**, ce qui est relativement correct pour un problème déséquilibré.
- L'**écart-type faible (0.01)** suggère une bonne stabilité du modèle à travers les différents folds de la validation croisée.

Conclusion: L'intégration de SMOTE dans la pipeline de validation croisée contribue à améliorer les performances de détection des cas positifs, ce qui est essentiel dans le cadre d'une application médicale. Ce **modèle équilibré, stable et régularisé** constitue une **base solide pour prédire le risque de cancer**, en particulier lorsque l'objectif est de **minimiser les faux négatifs**, plus graves que les fausses alertes dans ce contexte.

Il offre un bon compromis entre sensibilité et précision, et peut servir de **fondation pour des modèles plus complexes ou des systèmes d'aide à la décision clinique**.



- **438.8 vrais négatifs (TN)** : le modèle prédit correctement la majorité des personnes sans cancer.
- **16.4 faux positifs (FP)** : le modèle prédit "cancer" à tort pour des personnes qui ne l'ont pas.
- **15.0 vrais positifs (TP)** : le modèle détecte une partie des personnes atteintes de cancer.
- **67.4 faux négatifs (FN)** : le modèle **rate encore une grande partie des vrais cas de cancer**, ce qui est préoccupant dans un contexte médical.

Conclusion :

- ✓ Le modèle est **très bon pour détecter les non-malades**,
 ✗ mais **sous-performe fortement pour identifier les malades**, même après validation croisée.

3. Comparaison des approches d'équilibrage et de validation pour le modèle Random Forest

Au cours de nos expérimentations, nous avons évalué plusieurs stratégies visant à compenser le déséquilibre des classes (faible proportion de cas positifs) tout en testant différentes méthodes d'évaluation. L'objectif était d'identifier la configuration offrant le meilleur compromis entre **précision globale, capacité de détection des cas positifs et stabilité des performances**.

a. Modèle sans équilibrage (dataset déséquilibré)

Dans cette configuration, le modèle est entraîné sur un jeu de données fortement déséquilibré, avec une majorité d'individus n'ayant pas eu de cancer.

- **Accuracy élevée (~0.85)** : le modèle semble performant globalement.
- **Mais recall très faible (~0.18)** : il échoue à détecter la majorité des cas positifs.

Ce type de modèle est peu exploitable dans un cadre médical, car il néglige les individus les plus à risque.

b. Équilibrage par undersampling

Un sous-échantillonnage de la classe majoritaire permet de rétablir un équilibre entre les deux classes, en réduisant artificiellement le nombre de cas négatifs.

- **Accuracy : ~0.71**
- **Recall (classe cancer) : ~0.73**
- **F1-score : ~0.70**

Le modèle devient beaucoup plus équilibré, au prix d'une légère perte d'information due à la réduction de l'échantillon d'entraînement.

c. Équilibrage par SMOTE (oversampling synthétique)

Avec SMOTE, nous générions de manière synthétique de nouveaux exemples dans la classe minoritaire, sans suppression de données existantes.

- **Accuracy : ~0.84**
- **Recall (classe cancer) : ~0.56**

L'algorithme détecte davantage de cas positifs qu'en version brute, mais reste en deçà des performances obtenues avec undersampling.

d. SMOTE combiné à une validation croisée

La méthode la plus robuste consiste à intégrer SMOTE dans une pipeline de **validation croisée à 5 folds**, afin de mieux généraliser les performances du modèle.

- **Accuracy moyenne : ~0.84**
- **Recall : ~0.566**
- **Précision : ~0.655**
- **F1-score : ~0.577**
- **Écart-type faible (~0.01)** : performances stables d'un fold à l'autre.

Ce pipeline propose un bon compromis entre détection des cas positifs et précision globale, tout en assurant une bonne stabilité.

Conclusion générale

Parmi les différentes configurations testées, l'approche la plus équilibrée reste **Random Forest avec validation croisée et SMOTE**.

Elle permet de maximiser la **sensibilité du modèle** (rappel) tout en maintenant des niveaux de précision satisfaisants. Dans un contexte médical, cette stratégie est particulièrement pertinente car elle **réduit le risque de faux négatifs**, ce qui est crucial pour le dépistage et la prévention.

Bien que cela puisse entraîner un léger excès de faux positifs, ce compromis reste acceptable et conforme aux exigences d'une application en santé publique.

Approche	Méthode d'équilibrage	Accurac y	Recall (Classe 1)	Précision (Classe 1)	F1-score (Classe 1)	Écart-type Accuracy	Remarques
Modèle de base (déséquilibré)	Aucun	0.85	0.18	0.44	0.25	-	Très biaisé, ignore les cas positifs
Undersampling	Sous-échantillonnage	0.71	0.73	0.68	0.70	~0.03	Bon équilibre, mais perte de données
Validation croisée + Undersampling	Sous-échantillonnage	0.706	0.707	0.708	0.706	0.031	Stable, efficace, mais basé sur moins d'échantillons
SMOTE uniquement	Sur-échantillonnage	0.838	0.55	0.616	0.57	-	Moins biaisé, plus équilibré
Validation croisée + SMOTE	Sur-échantillonnage	0.84	0.566	0.655	0.577	0.01	Meilleur compromis stabilité/qualité

Cette configuration offre des performances équilibrées ($f1\text{-score} \approx 0.71$, $recall \approx 0.73$), avec une bonne stabilité. C'est actuellement la meilleure combinaison testée.

Pour comparer les performances, nous testons également un second type de modèle : la régression logistique. Ce modèle linéaire est souvent plus interprétable que les forêts aléatoires, ce qui peut être un atout en contexte médical.

E. Régression Logistiques: Comparaison avec une modèle linéaire

1. Évaluation du modèle : Régression Logistique + Undersampling

Résultat :

Logistic Regression
Accuracy moyenne : 0.693
Recall moyen : 0.694
Précision moyenne : 0.679

Interprétation des résultats

L'approche par régression logistique, combinée à un **undersampling de la classe majoritaire**, permet d'obtenir des performances à la fois **simples et équilibrées** :

- **Recall : 0.694**
Le modèle parvient à détecter environ **69 % des cas de cancer**, soit une sensibilité bien supérieure à celle du modèle Random Forest non équilibré (recall ≈ 0.18). Cela montre une capacité réelle à identifier les cas positifs.
- **Précision : 0.679**
Environ **68 % des prédictions positives** sont correctes, ce qui est tout à fait acceptable dans un contexte médical, où la qualité des alertes compte autant que leur nombre.
- **Accuracy : 0.693**
Bien que légèrement inférieure à celle des modèles Random Forest (~ 0.84), cette performance globale reste solide, notamment compte tenu de la balance entre les classes.

Conclusion :

La **régression logistique avec undersampling** s'impose comme une solution **simple, efficace et interprétable**. Elle offre un **bon compromis entre précision et recall**, deux éléments essentiels dans le domaine de la santé publique où **minimiser les faux négatifs** est souvent plus critique que maximiser l'accuracy.

À l'inverse, l'utilisation de SMOTE avec ce même algorithme a conduit à une **baisse marquée de la précision (≈ 0.29)**, entraînant une surdétection des cas positifs (faux positifs trop nombreux). Cela traduit une perte de fiabilité pour une application réelle.

En résumé, bien que la régression logistique soit légèrement moins performante que la Random Forest en termes d'accuracy brute, elle propose une **meilleure stabilité entre sensibilité et précision**, ce qui en fait un **excellent modèle de référence**, surtout dans un contexte à forte implication clinique.

2. Évaluation du modèle : Régression Logistique + SMOTE

Résultat :

Logistic Regression
Accuracy moyenne : 0.696
Recall moyen : 0.709
Précision moyenne : 0.295

Interprétation des résultats

- **Recall : 0.709**
Le modèle détecte environ **71 % des cas de cancer**, ce qui témoigne d'une **bonne sensibilité**. Cela en fait un outil pertinent pour **minimiser les cas non détectés**, ce qui est particulièrement crucial en santé publique.
- **Précision : 0.295**
Toutefois, près de **70 % des individus prédisits comme malades ne le sont pas réellement** (faux positifs). Ce taux élevé pourrait générer un nombre important **d'alertes inutiles**, avec un risque de sur-solicitation du système médical.
- **Accuracy : 0.696**
La précision globale du modèle reste correcte, et montre qu'il parvient à apprendre malgré le déséquilibre initial.

Conclusion

L'utilisation de **SMOTE avec la régression logistique** permet d'**améliorer sensiblement la capacité du modèle à détecter les cas positifs** (hausse du recall), mais au **prix d'une baisse significative de la précision**.

Dans un **contexte médical**, ce compromis peut être **acceptable**, dans la mesure où **il vaut mieux détecter à tort qu'ignorer un véritable cas de cancer**. Néanmoins, un tel modèle nécessiterait d'être **accompagné de mesures complémentaires** (tests de confirmation, consultation médicale...) pour **filtrer les faux positifs**.

En résumé, cette approche met l'accent sur la **sensibilité**, mais sa faible précision limite son usage autonome. Elle peut cependant jouer un rôle de **premier filtre efficace**, à intégrer dans une stratégie de dépistage plus complète.

F. Analyse comparative des modèles de classification

Dans le cadre de notre projet de prédiction du risque de cancer, plusieurs modèles de classification supervisée ont été testés — principalement **Random Forest** et **Régression Logistique**— afin d'identifier la configuration offrant le meilleur compromis entre **précision**, **recall** (**sensibilité**), **f1-score** et **stabilité**.

1. Déséquilibre des classes et stratégies de correction

Le jeu de données initial présente un fort déséquilibre : les personnes ayant déjà eu un cancer (classe positive) sont nettement moins nombreuses que celles qui n'en ont jamais eu (classe négative). Ce déséquilibre peut fausser les prédictions, car les modèles ont tendance à privilégier la classe majoritaire.

Deux méthodes principales ont été mises en œuvre pour corriger ce biais :

- **Undersampling** : réduction du nombre d'exemples dans la classe majoritaire.

- **SMOTE (Synthetic Minority Over-sampling Technique)** : génération de nouveaux exemples synthétiques dans la classe minoritaire.

2. Méthodologie d'évaluation

Chaque modèle a été testé dans plusieurs combinaisons, avec ou sans équilibrage, et avec ou sans **validation croisée à 5 folds**.

Les performances ont été évaluées selon les indicateurs suivants :

- **Accuracy** : proportion globale de bonnes prédictions.
- **Recall** : capacité à bien identifier les cas de cancer.
- **Précision** : part de vraies alertes parmi toutes les alertes positives.
- **F1-score** : équilibre entre précision et rappel.
-

3. Résultats clés

Random Forest

- **Sans SMOTE (undersampling uniquement)** :

Très bonne accuracy (jusqu'à 0.85), mais **recall très faible (0.18)**, ce qui signifie que le modèle détecte mal les cas positifs.
- **Avec SMOTE** :

Le recall s'améliore nettement (≈ 0.55), mais on observe une **légère baisse de précision**. Cela montre une meilleure détection des cas positifs, mais au prix de plus de faux positifs.
- **Validation croisée + undersampling** :

Les scores sont **très équilibrés** : accuracy ≈ 0.71 , recall ≈ 0.73 , f1-score ≈ 0.71 . C'est la configuration **la plus stable**.
- **Validation croisée + SMOTE** :

Accuracy ≈ 0.84 , mais recall et f1-score diminuent légèrement, ce qui peut indiquer une instabilité dans la détection des cas rares.

Régression Logistique

- **Avec undersampling** :

Résultats équilibrés : accuracy ≈ 0.693 , recall ≈ 0.694 , précision ≈ 0.679 . Le modèle reste cohérent.
- **Avec SMOTE** :

Le recall monte à 0.709, mais la précision chute drastiquement à 0.295. Le modèle devient trop sensible et produit **trop de fausses alertes**.

Conclusion:

Les différentes expérimentations montrent que le modèle **Random Forest, combiné à l'undersampling et évalué par validation croisée**, fournit **le meilleur compromis** entre les principales métriques. Il détecte efficacement les cas positifs (rappel élevé), tout en limitant les faux positifs (bonne précision), et affiche une stabilité appréciable sur les folds de validation.

Recommandation pour la mise en production :
Nous recommandons l'utilisation du **modèle Random Forest + undersampling + validation**

croisée. Ce choix garantit un modèle **fiable, équilibré et robuste**, capable d'être intégré dans une solution automatisée d'aide à la détection du risque de cancer.

G. Modèle Random Forest avec undersampling et validation croisée

Après avoir testé plusieurs combinaisons de modèles et de techniques d'équilibrage, nous pouvons désormais comparer leurs performances à l'aide des métriques clés (accuracy, recall, f1-score, etc.). L'objectif est d'identifier la meilleure approche pour une éventuelle mise en production.

Résultat :

Résultats (Random Forest + undersampling + CV) :

Accuracy : 0.708 (± 0.027)
Recall : 0.757 (± 0.035)
Precision : 0.692 (± 0.035)
F1 : 0.722 (± 0.018)

Interprétation :

- **Recall élevé (0.757)** : Le modèle identifie bien les cas positifs (personnes ayant eu un cancer).
- **Bonne précision (0.692)** : Relativement peu de faux positifs.
- **F1-score équilibré (0.722)** : Bon compromis entre recall et précision.
- **Écart-types faibles** : Le modèle est **stable** sur les différentes itérations de la validation croisée.

Pour une application en production, nous recommandons d'utiliser une **Random Forest avec undersampling et une évaluation par validation croisée**.

Ce modèle offre un bon équilibre entre **précision** et **rappel**, ce qui est essentiel dans un contexte médical où il est important d'identifier un maximum de cas positifs tout en évitant un grand nombre de fausses alertes.

De plus, sa stabilité statistique (faibles écarts-types) en fait un candidat robuste pour une mise en production fiable.

Conclusion Général:

Au terme de notre étude comparative, plusieurs modèles de classification ont été testés afin de prédire si un individu a déjà eu un cancer, en tenant compte des déséquilibres présents dans les données.

Les différentes approches (Random Forest et Régression Logistique) ont été évaluées avec deux méthodes d'équilibrage : **undersampling** et **SMOTE**, associées ou non à de la **validation croisée**. Ces tests ont permis de mieux comprendre l'impact des choix de prétraitement et d'équilibrage sur les performances des modèles.

Les résultats montrent que le modèle **Random Forest avec undersampling et validation croisée** présente un bon compromis entre **rappel élevé** (bonne détection des cas positifs), **précision acceptable** (peu de fausses alertes) et **stabilité des résultats**. Il apparaît ainsi comme le modèle le plus fiable et équilibré pour une éventuelle mise en production.

Nous recommandons ce modèle comme solution par défaut pour un système automatisé de détection de risque de cancer, en raison de sa robustesse, de sa capacité à gérer les données déséquilibrées et de ses performances globales solides.

Random Forest

Méthode	Accuracy	Recall (classe 1)	Précision	F1-score	Remarques
Sans équilibrage	~0.85	~0.18	Élevée	Faible	Très mauvais pour détecter les cas positifs.
Avec SMOTE	~0.84	~0.55	Moyenne	~0.57	Meilleure détection, mais plus de faux positifs.
Undersampling + CV	~0.71	~0.73	~0.70	~0.71	Excellent compromis, modèle stable.
SMOTE + CV	~0.84	~0.56	~0.66	~0.57	Bon, mais moins stable et recall plus bas.

Régression Logistique

Méthode	Accuracy	Recall (classe 1)	Précision	F1-score	Remarques
Undersampling	~0.69	~0.69	~0.68	~0.69	Modèle cohérent, bon rappel et précision.
SMOTE	~0.69	~0.71	~0.30	~0.42	Trop de faux positifs, précision faible.