Rapport sur :

Prédiction de l'Espérence de vie Au MAROC :

« 2020-2030 »

(Maching Learning)

-Présenté par :

-MOHAMED AMHAL

-Encadré par :

-Prof :YASSINE.ALAMRANI

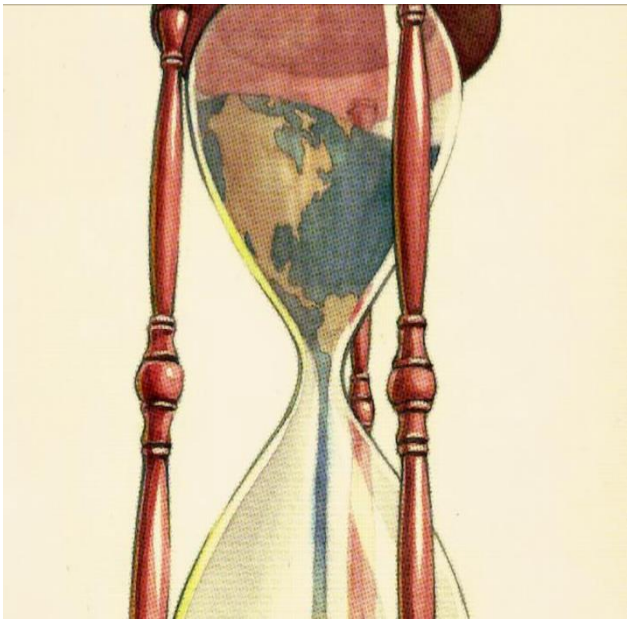
Date : Le 02/01/2024

Le sommaire :

I . Introduction :

L'espérance de vie, en tant qu'indicateur clé de la santé d'une population, revêt une importance cruciale dans l'évaluation du bien-être d'une société. Au Maroc, la dynamique démographique et les avancées en matière de santé au cours de la dernière décennie ont suscité un intérêt particulier quant à l'évolution future de l'espérance de vie.

Ce rapport propose une analyse prospective de l'espérance de vie au Maroc pour la période 2020-2030, en se basant sur des techniques avancées de data mining.



Le Maroc, pays d'Afrique du Nord, a connu des Changements significatifs dans divers domaines tels que la santé, l'éducation, l'économie et la technologie. Ces transformations influent directement sur la santé de sa population et, par conséquent, sur l'espérance de vie. À travers ce rapport, nous cherchons à comprendre les tendances passées, à anticiper les évolutions futures et à identifier les principaux facteurs qui façonnent l'espérance de vie au Maroc.

Avant d'entamer une analyse ciblée sur l'espérance de vie au Maroc pour la décennie à venir, il est impératif de contextualiser nos travaux en réalisant une analyse rétrospective à l'échelle mondiale. Pour ce faire, nous avons rassemblé des données exhaustives couvrant la période allant de l'an 2000 à 2015, provenant de sources diverses et représentatives de pays du monde entier. Cette approche globale vise à appréhender les tendances universelles en matière de santé et de longévité, établissant ainsi un référentiel comparatif pour les futurs développements au Maroc.

Au cours de cette phase initiale, nous explorons les évolutions majeures dans les domaines de la médecine, de la technologie, de l'éducation et des conditions socio-économiques qui ont marqué cette période à l'échelle mondiale. Cette contextualisation mondiale nous permettra de discerner des schémas et des déterminants communs qui ont pu influencer la longévité de populations diverses. En outre, elle offre une opportunité unique de tirer des leçons des succès et des défis rencontrés par d'autres nations dans leur quête d'amélioration de l'espérance de vie.

Fort de cette compréhension mondiale, notre objectif est de canaliser ces enseignements pour prédire avec précision l'évolution de l'espérance de vie spécifiquement au Maroc pour la période allant de 2020 à 2030. En combinant les données rétrospectives mondiales avec les particularités propres au contexte marocain, nous visons à élaborer des modèles robustes capables d'anticiper les trajectoires futures de la longévité dans cette région du monde, tout en tenant compte des facteurs socio-économiques, sanitaires et environnementaux spécifiques au pays. Cette approche holistique assure une fondation solide pour des prédictions informées et éclairées sur l'espérance de vie au Maroc au cours de la prochaine décennie.

Ce projet de Data Mining visant à analyser l'espérance de vie au Maroc pour la période 2020-2030 est structuré en plusieurs phases méthodiques, chacune contribuant de manière significative à la compréhension globale de cette dynamique complexe. Le plan du projet est articulé autour des étapes suivantes :

1- **Compréhension des Données :** Présentation des différentes sources de données utilisées, la justification du choix de ces données, et la définition détaillée de chaque variable.
2- **Data cleaning :** (nettoyage des données) améliorer la qualité des données en garantissant leur exactitude, leur cohérence et leur fiabilité.
3- **Analyse Exploratoire :** Une description des principaux résultats de l'analyse exploratoire, notamment les tendances, les corrélations, et les insights découverts à travers la visualisation des données.
4- **Choix des Modèles de Machine Learning :** Une justification détaillée des modèles sélectionnés.
5- **Comparaison des Modèles :** Une évaluation approfondie des performances de chaque modèle, avec des métriques spécifiques utilisées pour la comparaison.
6- **Le déploiement du modèle choisi** : la conception et le développement de l'application web, les choix technologiques, et la manière dont elle interagit avec les modèles de machine Learning.

I. Compréhension des Données :

La phase de Compréhension des Données constitue le socle fondamental de toute entreprise d'analyse, dévoilant la quintessence des informations à notre disposition. Au cœur de cette étape, nous plongeons dans l'univers complexe des données, cherchant à saisir la nature, la structure et les particularités de chaque variable. Cette exploration initiale est cruciale pour établir une base solide, garantissant une interprétation éclairée des résultats qui découleront de notre analyse. Au fil de cette section, nous détaillerons les sources de données exploitées, clarifierons les définitions de chaque variable, et mettrons en lumière les choix stratégiques qui ont façonné la trajectoire de notre projet. La Compréhension des Données, par sa nature exploratoire, pave la voie à une analyse rigoureuse et éclairée, jetant ainsi les bases d'une compréhension approfondie des mécanismes sous-jacents à l'espérance de vie au Maroc.

1-la base de données :

Cette base de données constitue une précieuse compilation de données mondiales, embrassant la période de 2000 à 2015, et offre une vue exhaustive sur divers aspects liés à la santé et au bien-être dans chaque pays. Chaque entrée dans cette base représente un individu pour une année donnée, avec un total de 15 lignes par pays, correspondant à chaque année de la période étudiée. Les variables indicatrices, soigneusement sélectionnées, offrent une perspective holistique, couvrant des éléments tels que la couverture vaccinale (Hépatite B, Rougeole, Polio, Diphtérie), la prévalence du VIH/SIDA, la mortalité infantile, les dépenses totales de santé, le Produit Intérieur Brut (PIB), l'Indice de Masse Corporelle (IMC), la consommation d'alcool, le niveau de scolarité, et bien sûr, l'espérance de vie.

Chaque variable représente une facette unique et cruciale pour la compréhension des dynamiques de santé à l'échelle mondiale. Le statut socio-économique, symbolisé par la variable "Statut", offre une distinction pertinente entre les pays développés et en développement. Cette richesse d'informations, combinée à la diversité des indicateurs de santé, crée une toile complexe mais essentielle pour l'analyse approfondie de l'espérance de vie. Ainsi, cette base de données offre une opportunité inestimable d'exploration, permettant une compréhension détaillée des facteurs qui sous-tendent les disparités en matière de santé et d'espérance de vie à travers le globe.

- Voici un instantané captivant de notre base de données

Index	pays	Année	Statut	Population	Hépatite_B	Rougeole	Polio	Diphtérie	VIH/SIDA	rtalité_infan	cès_des_moins_de_5_a	Dépenses_totales	PIB	IMC	Maigreur_1_19_an	Alcool	Scolarité	Espérance_de_vie
1680	Morocco	2015	Developing	3.48332e+06	99	17	99	99	0.1	17	20	nan	2847.29	58.5	6.4	nan	12.1	74.3
1681	Morocco	2014	Developing	3.43188e+06	99	10	99	99	0.1	18	21	5.91	3154.51	57.5	6.4	0.43	12.1	74.1
1682	Morocco	2013	Developing	3.38248e+07	99	92	99	99	0.1	18	21	5.94	3111.76	56.5	6.4	0.45	12.1	73.9
1683	Morocco	2012	Developing	3.33338e+07	99	668	99	99	0.1	19	22	6.15	294.747	55.5	6.3	0.55	11.6	73.6
1684	Morocco	2011	Developing	3.28588e+07	98	982	98	99	0.1	19	22	5.99	339.916	54.6	6.3	0.54	11.2	73.3
1685	Morocco	2010	Developing	3.24964e+06	98	633	99	99	0.1	20	23	5.86	2834.25	53.6	6.3	0.56	10.7	72.8
1686	Morocco	2009	Developing	3.19899e+07	98	834	99	99	0.1	20	23	5.67	2861.55	52.7	6.4	0.62	10.5	72.3
1687	Morocco	2008	Developing	3.15969e+07	97	1455	99	99	0.1	21	24	5.41	2884.95	51.7	6.4	0.51	10.3	71.8
1688	Morocco	2007	Developing	3.12259e+07	95	2248	95	95	0.1	21	25	5.48	2494.35	5.8	6.4	0.56	10	71.4
1689	Morocco	2006	Developing	3.86935e+06	95	1217	97	97	0.1	22	25	5.23	2191.48	49.9	6.4	0.58	10	71
1690	Morocco	2005	Developing	35217	96	0	98	98	0.1	22	26	5.6	213.756	49.1	6.5	0.47	9.8	77
1691	Morocco	2004	Developing	3.17928e+06	95	6399	97	97	0.1	23	27	5.22	1948.81	48.2	6.5	0.56	9.6	72
1692	Morocco	2003	Developing	2.98439e+07	9	10841	91	91	0.1	24	28	5.25	1721.97	47.3	6.6	0.58	9.3	69.9
1693	Morocco	2002	Developing	2.95124e+07	92	6000	94	94	0.1	25	29	5.31	1413.76	46.5	6.6	0.46	8.8	69.5
1694	Morocco	2001	Developing	2.91818e+07	84	2724	93	96	0.1	26	30	4.44	1336.78	45.7	6.7	0.46	8.5	69
1695	Morocco	2000	Developing	2.88496e+07	43	7368	95	95	0.1	27	32	4.18	1332.38	44.8	6.7	0.45	8	68.6
1696	Mozambique	2015	Developing	281691	8	79	8	8	3.9	60	81	nan	528.313	22.6	3.6	nan	9.1	57.6
1697	Mozambique	2014	Developing	2.72124e+07	79	9	79	79	4.1	61	84	6.98	623.287	22.2	3.6	0.01	9.1	56.7
1698	Mozambique	2013	Developing	2.64344e+07	78	8	78	78	5.1	62	87	5.9	65.9857	21.8	3.6	1.16	9.1	55.3
1699	Mozambique	2012	Developing	2.56767e+06	76	145	73	76	6.9	64	90	5.58	566.514	21.3	3.6	1.19	9.2	54.8
1700	Mozambique	2011	Developing	249395	76	177	73	76	9.6	66	94	6.23	526.531	2.9	3.7	0.94	9.5	54.3
1701	Mozambique	2010	Developing	2.42214e+06	74	2321	73	74	10.8	69	98	5.38	419.226	2.5	3.7	0.96	9.3	54

2-les définitions des variables :

-Pays :

=>Cette variable représente le nom du pays auquel chaque individu appartient dans la base de données.

=>Elle offre une dimension géographique essentielle, permettant de segmenter et d'analyser les données en fonction des spécificités propres à chaque nation.

-Année :

=>L'année indique la période à laquelle les données ont été recueillies, couvrant la plage temporelle de 2000 à 2015.

=>C'est une variable temporelle cruciale qui permet d'observer les évolutions au fil du temps et de repérer les tendances émergentes dans la santé et l'espérance de vie.

-Statut :

=>Le statut différencie les pays en deux catégories : "Développé" ou "En développement".

=>Cette variable socio-économique offre un éclairage sur les disparités de développement entre les nations, jouant un rôle clé dans l'analyse des différences d'espérance de vie.

-Population :

=>La population représente le nombre d'individus dans chaque pays pour une année spécifique.

=>Elle permet d'appréhender l'ampleur démographique de chaque contexte, influençant potentiellement les statistiques de santé et d'espérance de vie.

-Hépatite_B :

=>Indiquant le pourcentage de couverture vaccinale contre l'hépatite B dans chaque pays, cette variable mesure l'impact des programmes de vaccination.

=>Elle joue un rôle crucial dans l'évaluation de la santé publique et des risques de maladies hépatiques.

-Rougeole :

=>Cette variable représente la couverture vaccinale contre la rougeole, offrant des indications sur la prévalence de cette maladie évitable par la vaccination.

=>Elle est pertinente pour évaluer l'efficacité des programmes de vaccination infantile.

-Polio :

=>Mesurant la couverture vaccinale contre la poliomyélite, cette variable souligne l'effort pour éradiquer cette maladie.

=>Elle offre des insights sur la robustesse des systèmes de santé et les succès des campagnes de vaccination.

-Diphtérie :

=>La couverture vaccinale contre la diphtérie évalue la protection contre cette maladie potentiellement mortelle.

=>Cette variable contribue à évaluer l'efficacité des programmes de vaccination infantile.

-VIH/SIDA :

=>Indiquant la prévalence du VIH/SIDA dans chaque pays, cette variable met en lumière la charge de cette maladie et son impact sur l'espérance de vie.

=>Elle offre un aperçu crucial des défis de santé publique liés au VIH/SIDA.

-Mortalité Infantile :

=>Mesurant le nombre d'enfants décédés avant leur premier anniversaire par 1 000 naissances vivantes, cette variable reflète la santé infantile.

=>Elle est un indicateur significatif du niveau de soins de santé maternelle et infantile dans chaque pays.

-Décès des Moins de 5 Ans :

=>Représentant le nombre d'enfants décédés avant l'âge de 5 ans pour 1 000 naissances vivantes, cette variable évalue la mortalité infantile et juvénile globale.

=>Elle offre des informations cruciales sur la santé des enfants à un stade précoce de leur vie.

-Dépenses Totales :

=>Cette variable indique les dépenses totales de santé par habitant dans chaque pays, reflétant l'investissement dans les systèmes de santé.

=>Elle influence directement la qualité des soins de santé accessibles à la population.

-PIB :

=>Mesurant le Produit Intérieur Brut par habitant, le PIB est un indicateur économique qui influence les conditions de vie et de santé.

=>Il offre une perspective sur le niveau de développement économique des pays, impactant potentiellement l'espérance de vie.

-IMC (Indice de Masse Corporelle) :

=>L'IMC évalue le poids relatif à la taille, fournissant des indications sur la nutrition et la santé générale.

=>Cette variable est liée à des implications directes sur la prévalence des maladies liées au poids et à l'espérance de vie.

-Maigreur 1-19 Ans :

=>Indiquant la proportion d'enfants de 1 à 19 ans souffrant de maigreur, cette variable offre des insights sur la malnutrition infantile.

=>Elle est pertinente pour évaluer la qualité de la nutrition infantile et ses conséquences sur la santé à long terme.

-Alcool :

=>Représentant la consommation d'alcool par habitant, cette variable souligne les habitudes de consommation et les risques associés à la santé.

=>Elle peut influencer directement la prévalence des maladies liées à l'alcool et donc l'espérance de vie.

-Scolarité :

=>Mesurant la moyenne d'années de scolarité dans chaque pays, cette variable évalue le niveau d'éducation de la population.

=>Elle est un indicateur socio-économique essentiel, influençant divers aspects de la vie, y compris la santé.

-Espérance de Vie :

=>L'espérance de vie représente la durée moyenne qu'un individu peut s'attendre à vivre dans un pays donné.

=>Cette variable est le point central de l'analyse, synthétisant l'impact cumulatif de toutes les variables précédentes sur la santé et la longévité de la population.

II. Prétraitement des données :

Le data cleaning, également appelé "nettoyage des données", est le processus consistant à identifier, corriger ou supprimer les erreurs, les incohérences et les anomalies présentes dans un ensemble de données. L'objectif principal du data cleaning est d'améliorer la qualité des données en garantissant leur exactitude, leur cohérence et leur fiabilité. Ce processus implique généralement la détection et la correction des valeurs manquantes, des erreurs de saisie, des duplicatas, des valeurs aberrantes (outliers), ainsi que d'autres anomalies qui pourraient compromettre l'intégrité et la pertinence des données. Le data cleaning est une étape essentielle dans le prétraitement des données avant toute analyse ou application de modèles, car des données propres et de haute qualité sont cruciales pour obtenir des résultats fiables et significatifs.

1-Importation des données csv dans Python :

L'étape d'importation des données CSV dans Python est cruciale pour amorcer notre exploration et notre analyse de la base de données. À travers l'utilisation de bibliothèques telles que **Pandas**, nous avons orchestré un processus fluide et efficace d'importation des données, transformant le fichier CSV en une structure de données manipulable en Python. Cette phase préliminaire permet d'établir une connexion essentielle entre notre environnement de développement et le jeu de données, préparant ainsi le terrain pour une analyse approfondie. Les fonctionnalités puissantes de Pandas nous offrent la flexibilité nécessaire pour explorer, filtrer et manipuler les données avec aisance, jetant ainsi les bases d'une exploration des variables et des tendances à venir. L'importation des données est ainsi le premier pas vers la découverte et la compréhension des informations encapsulées dans la base de données, inaugurant notre parcours analytique au cœur de l'univers des données de santé et d'espérance de vie.

```
import pandas as pd
```

```
    #importer la base de données :
df = pd.read_csv(r"C:\Users\Surface\OneDrive\Bureau\life_expectancy.csv",encoding='ISO-8859-1')

    #afficher les informations de chaque colonne:
```

-le code **« df.info () »** permet d'afficher les informations de chaque variable :

```
#   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
0   pays                      2848 non-null   object
1   Année                     2848 non-null   int64
2   Statut                    2848 non-null   object
3   Population                2204 non-null   float64
4   Hépatite_B                2306 non-null   float64
5   Rougeole                  2848 non-null   int64
6   Polio                     2829 non-null   float64
7   Diphtérie                 2829 non-null   float64
8   VIH/SIDA                  2848 non-null   float64
9   Mortalité_infantile       2848 non-null   int64
10  Décès_des_moins_de_5 ans  2848 non-null   int64
11  Dépenses_totales          2627 non-null   float64
12  PIB                       2406 non-null   float64
13  IMC                       2816 non-null   float64
14  Maigreur_1_19_ans         2816 non-null   float64
15  Alcool                    2660 non-null   float64
16  Scolarité                 2688 non-null   float64
17  Espérance_de_vie          2848 non-null   float64
```

2-les statistiques des variables qualitatifs et quantitatifs :

Le code :

```
      #voir les statistiques des variables quantitatifs:
stat_quantita = df.describe()

      #voir les statistiques des variables qualitatifs:
stat_qualita = df.describe(include='object')

      #voir le nombre des valeurs manquantes de chaque varaible :
valeur_manqq = df.isnull().sum().sort_values(ascending = False)
```

-les statistique qualitatifs :

Index	pays	Statut
count	2848	2848
unique	178	2
top	Afghanistan	Developing
freq	16	2352

-les statistique quantitatifs :

Index	Année	Population	Hépatite_B	Rougeole	Polio	Diphtérie	VIH/SIDA	Mortalité_infantile	s_des_moins_de_5	Dépenses_totales	PIB	IMC	Maigreur_1_19_an	Alcool	Scolarité	Espérance_de_vie
count	2848	2204	2306	2848	2829	2829	2848	2848	2848	2627	2406	2816	2816	2660	2688	2848
mean	2007.5	1.28346e+07	81.0768	2083.08	82.6822	82.4514	1.75646	28.3599	39.5	5.93558	7664.4	38.5034	4.84723	4.63893	12.0602	69.3474
std	4.61058	6.19609e+07	25.0191	10249.1	23.435	23.6939	5.14894	117.188	159.801	2.50444	14466.2	19.9555	4.4437	4.06472	3.32016	9.52833
min	2000	34	1	0	3	2	0.1	0	0	0.37	1.68135	1	0.1	0.01	0	36.3
25%	2003.75	196758	77	0	78	78	0.1	0	0	4.24	477.542	19.5	1.6	0.93	10.2	63.5
50%	2007.5	1.39176e+06	92	16	93	93	0.1	3	4	5.76	1841.09	43.9	3.3	3.785	12.4	72.2
75%	2011.25	7.43895e+06	97	336.75	97	97	0.7	20	25	7.53	6265.66	56.2	7.125	7.81	14.3	75.8
max	2015	1.29386e+09	99	212183	99	99	50.6	1800	2500	17.6	119173	77.6	27.7	17.87	20.7	89

3-diviser la base de données et traitement des valeurs manquantes :

Avant la division de la base de données, on va afficher la somme des valeurs manquantes de chaque variable :

-le code :

```
#voir le nombre des valeurs manquantes de chaque varaible :
valeur_manqq = df.isnull().sum().sort_values(ascending = False)
```

-la visualisation :

Index	0
Population	644
Hépatite_B	542
PIB	442
Dépenses_totales	221
Alcool	188
Scolarité	160
Maigreur_1_19_ans	32
IMC	32
Polio	19
Diphtérie	19
pays	0
Mortalité_infantile	0
Décès_des_moins_de_5 ans	0
Année	0
VIH/SIDA	0
Rougeole	0
Statut	0
Espérance_de_vie	0

La division de la base de données en deux parties, qualitative et quantitative, s'est avérée être une stratégie judicieuse pour aborder efficacement les valeurs manquantes. Dans le cadre de la base de données, les variables qualitatives incluent des attributs tels que le pays, l'année, le statut, et d'autres indicateurs catégoriels, tandis que les variables quantitatives englobent des mesures telles que la population, les dépenses totales de santé, le PIB, et divers autres paramètres numériques.

```
      #diviser la base de donnes en deux types : qualitatif et quantitatif:
base_qualitatif = []
base_quantitatif = []
      #df.dtype => colonne,type de donnees enumerate =>pour avoir un dictionnaire
for i,j in enumerate(df.dtypes):
    if j == 'object':
        base_qualitatif.append(df.iloc[:,i]) #selectionner tt la colonne
    else:
        base_quantitatif.append(df.iloc[:,i])

      #transformer les listes en dataframe (transpose)
base_qualitatif = pd.DataFrame(base_qualitatif).transpose()
base_quantitatif = pd.DataFrame(base_quantitatif).transpose()
```

Pour traiter les valeurs manquantes dans **les variables qualitatives**, une approche consistait à remplacer les données manquantes par la modalité la plus fréquente de la variable respective (le mode). Cette méthode a été préférée pour préserver la distribution catégorielle originale des données, assurant ainsi une représentation fidèle de la variabilité dans ces caractéristiques qualitatives.

```
      #remplacer les valeurs manquantes des varaiables qualitatif (mode) :
base_qualitatif = base_qualitatif.apply(lambda x : x.fillna(x.value_counts().index[0]))
```

Quant aux variables quantitatives, l'approche adoptée a consisté à remplacer les valeurs manquantes par la moyenne arithmétique de la distribution correspondante (mean ()).

```
      #remplecer les valeurs manquantes des variables qualitatif:
base_quantitatif = base_quantitatif.apply(lambda x : x.fillna(x.mean()))
```

Cette segmentation entre variables qualitatives et quantitatives a permis une approche ciblée et spécifique à chaque type de données, assurant ainsi la cohérence et la qualité globale du jeu de données. En combinant une manipulation réfléchie des valeurs manquantes avec la préservation de la nature inhérente à chaque type de variable, cette démarche a jeté les bases d'une base

de données plus complète et prête à être explorée dans le cadre de notre analyse.

4-Autre techniques de nettoyages de données :

a-concaténer les mots des pays :

La gestion des noms de pays constitués de deux mots dans la base de données a été une étape importante pour assurer la cohérence et l'efficacité lors de l'entraînement des modèles. Étant donné que de nombreux algorithmes d'apprentissage automatique fonctionnent de manière optimale avec des données numériques ou textuelles uniformisées, nous avons entrepris une opération de prétraitement visant à concaténer les mots des noms de pays séparés par un espace, les remplaçant ainsi par un souligné ('_').

Cette transformation a permis de résoudre le défi des noms de pays composés, créant une convention unifiée pour représenter ces entités dans la base de données. Par exemple, un pays initialement enregistré comme "Afrique du Sud" a été transformé en "Afrique_du_Sud". Ce processus a été appliqué de manière systématique à l'ensemble de la base de données, assurant ainsi une homogénéité dans la représentation des noms de pays.

La concaténation des mots par un souligné a non seulement simplifié la gestion des données au cours de l'entraînement des modèles, mais a également contribué à éviter des erreurs potentielles liées à la reconnaissance des noms de pays lors des analyses futures. Cette adaptation a été documentée de manière transparente dans notre processus de prétraitement des données, garantissant la traçabilité et la compréhension de cette transformation dans le cadre de notre démarche analytique.

-le code :

```
    #concatenations des mots des pays :
base_qualitatif['pays'] = base_qualitatif['pays'].apply(lambda x: re.sub(r'[^a-zA-Z\s]', '',
                                                                         str(x)).replace(' ', '_') if x else '')
```

b-normaliser la variable ‘statut’ :

La normalisation de la variable "Statut" représente une étape cruciale dans notre processus de prétraitement des données, visant à rendre cette caractéristique plus adaptée aux algorithmes d'apprentissage automatique. Initialement, la variable "Statut" distinguait les pays entre "Developing" (en développement) et "Developed" (développé), offrant une perspective socio-économique importante pour l'analyse de l'espérance de vie.

Cependant, pour faciliter l'incorporation de cette information dans nos modèles, nous avons choisi de normaliser cette variable en utilisant une représentation binaire. Désormais, chaque pays est associé à une valeur binaire, où 0 représente la catégorie "Developing" et 1 correspond à la catégorie "Developed". Cette transformation simplifie le processus d'entraînement des modèles, lesquels interagissent plus aisément avec des variables binaires plutôt qu'avec des catégories textuelles.

Cette normalisation offre plusieurs avantages, notamment une réduction de la complexité du modèle, une accélération des calculs, et une interprétation plus directe des résultats. De plus, elle garantit une compatibilité optimale avec des algorithmes qui nécessitent des données numériques. Par exemple, un modèle pourra désormais attribuer des poids spécifiques à la variable "Statut" lors de son entraînement, facilitant ainsi la détection de corrélations et de tendances liées au développement socio-économique.

L'ensemble de ces ajustements a été méticuleusement documenté dans notre processus de prétraitement des données, contribuant à la transparence et à la reproductibilité de nos analyses futures. Cette normalisation, en renforçant la cohérence des données, s'inscrit dans notre démarche pour créer un ensemble de données homogène et optimisé pour l'application de modèles d'apprentissage automatique.

-le code :

```
    #transformer la variable categorique status en variable numerique :
    #Developing = 0 et  Developed = 1
base_qualitatif['Statut'] = df['Statut'].apply(lambda x: 0 if x == 'Developing' else 1)
```

À l'issue de ce processus de prétraitement où les bases de données qualitatives et quantitatives ont été manipulées de manière distincte pour répondre à leurs particularités respectives, une étape clé s'annonce : la concaténation de ces deux ensembles. Cette fusion des données qualitatives et quantitatives vise à créer un tableau de données unifié, intégrant les caractéristiques spécifiques de chaque type de variable. Cette convergence permettra de construire une représentation exhaustive de notre ensemble de données, prête à être explorée de manière approfondie au moyen d'analyses exploratoires, notamment la data visualisation.

III. Analyse Exploratoire :

L'analyse exploratoire joue un rôle essentiel dans la révélation désintrications subtiles et complexes inscrites au sein de notre ensemble de données. C'est une phase cruciale qui transcende la simple observation des variables pour devenir une exploration approfondie des relations, des modèles émergents, et des tendances significatives. En tant que fenêtre inaugurale sur les données, l'analyse exploratoire constitue un outil préliminaire puissant pour la découverte de nouveaux questionnements et la formulation d'hypothèses.

Son utilité réside dans la capacité à transformer des données brutes en connaissances exploitables. À travers des méthodes de visualisation sophistiquées, des statistiques descriptives, et des techniques de réduction dimensionnelle, l'analyse exploratoire offre une compréhension initiale des structures sous-jacentes dans notre base de données sur l'espérance de vie. Elle permet d'identifier des points saillants, de mettre en lumière des corrélations insoupçonnées, et d'orienter les investigations futures.

En révélant les intrications entre les variables, l'analyse exploratoire contribue à affiner nos hypothèses et à définir des axes d'approfondissement. C'est un prélude essentiel à des analyses plus avancées, orientant la démarche analytique tout en fournissant des repères cruciaux pour l'interprétation des résultats ultérieurs. En somme, l'analyse exploratoire se présente comme la passerelle inaugurale vers une compréhension approfondie et nuancée des mécanismes qui sous-tendent l'espérance de vie, orientant ainsi le cours de notre investigation.

Les bibliothèques utiles pour cette analyse exploratoire sont :

```
import matplotlib.pyplot as plt
import seaborn as sns
```

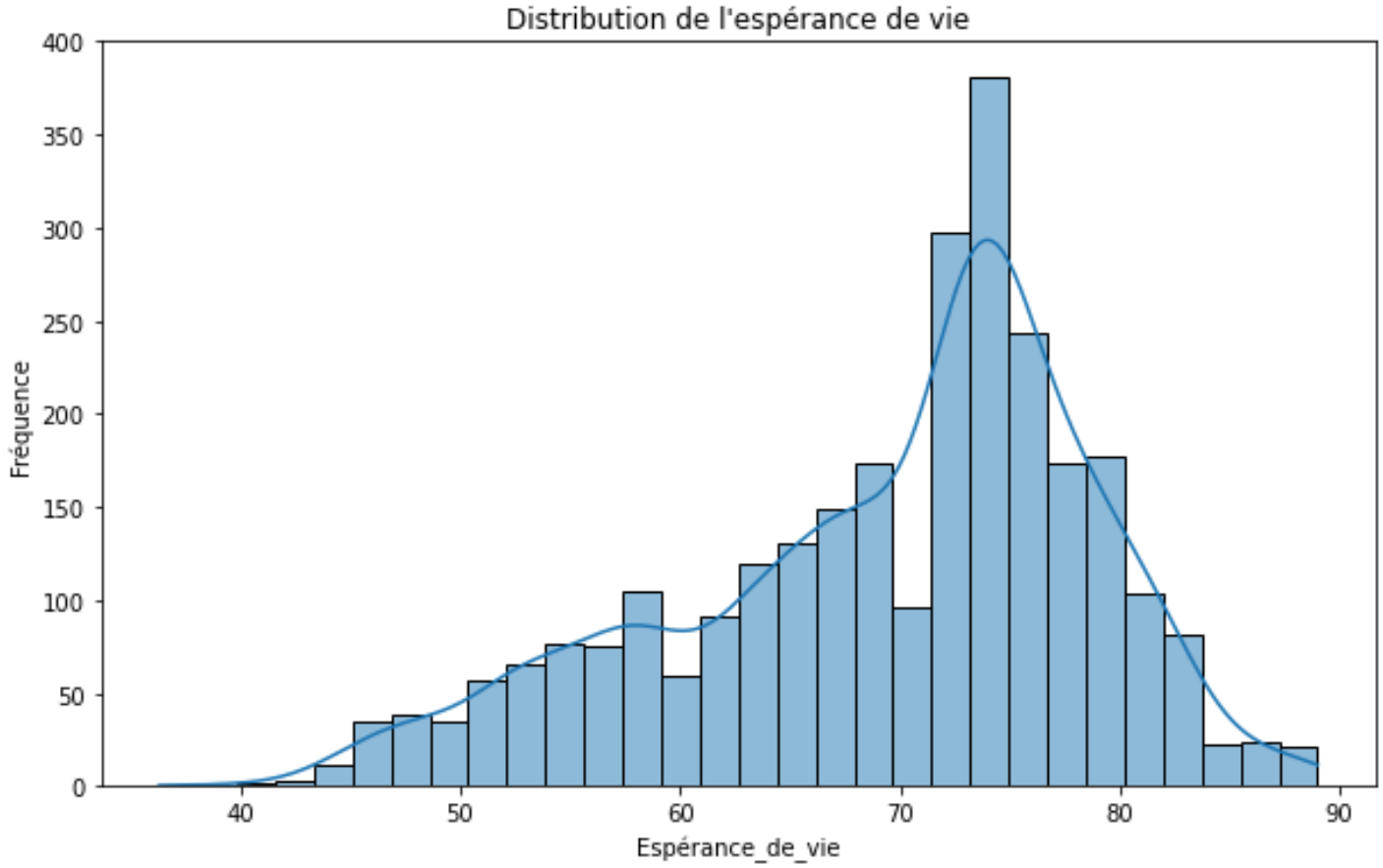
1-Histogramme de la variable « Espérance_de_vie » :

L'histogramme de la variable "Espérance_de_vie" offre une vision graphique captivante de la distribution de cette caractéristique clé au sein de notre ensemble de données sur l'espérance de vie. Cet outil visuel permet de saisir intuitivement la répartition des durées de vie dans la population étudiée. Chaque barre de l'histogramme représente une plage de valeurs d'espérance de vie, offrant une représentation visuelle de la fréquence d'occurrence de ces plages.

-le code de la visualisation :

```
    # Histogramme de la variable Y "Espérance_de_vie":

plt.figure(figsize=(10, 6))
sns.histplot(base['Espérance_de_vie'], bins=30, kde=True)
plt.title("Distribution de l'espérance de vie")
plt.xlabel('Espérance_de_vie')
plt.ylabel('Fréquence')
plt.show()
```

-L’histogramme :



=> On constate que la majorité de la population a une espérance de vie comprise entre 70 et 80 ans sur le graphique de l'histogramme.

2- Exploration des Relations entre les Variables Numériques à travers la Matrice de Corrélation :

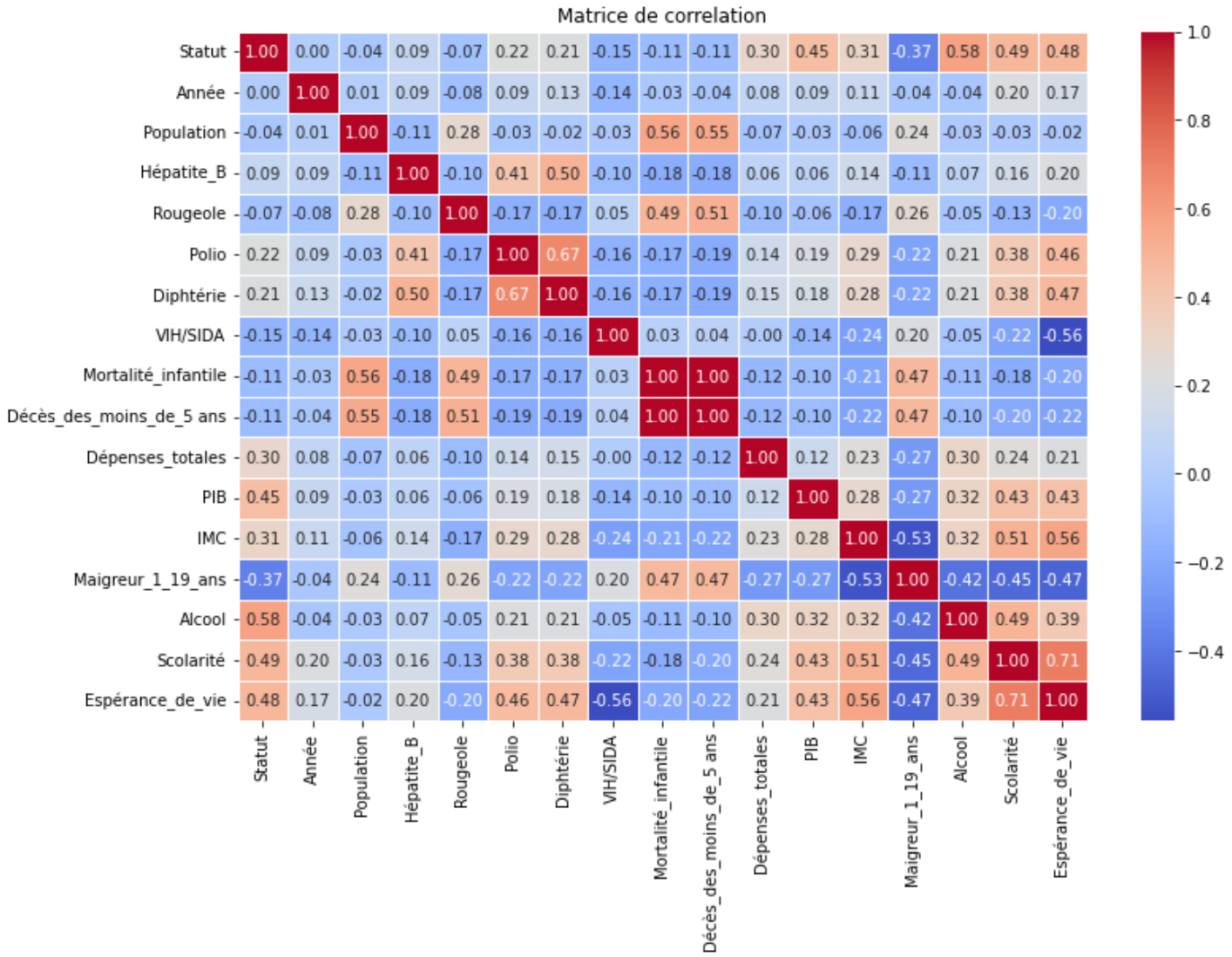
La matrice de corrélation entre les variables numériques constitue une fenêtre privilégiée pour sonder les liens intrinsèques entre différentes caractéristiques de notre ensemble de données. Cette représentation matricielle offre une vue d'ensemble des corrélations, mesurant la force et la direction des relations linéaires entre les variables. Les coefficients de corrélation, qui varient de -1 à 1, permettent d'appréhender la nature des interactions : une corrélation positive suggère une relation directe, tandis qu'une corrélation négative indique une relation inverse. Cette exploration approfondie des relations entre les

variables numériques jette les bases pour une compréhension plus nuancée des dynamiques à l'œuvre dans notre ensemble de données, guidant ainsi nos prochaines étapes d'analyse.

-le code :

```
#Matrice de correlation entre les variables numeriques :(voir les relation entre les variab

correlation_matrix = base.corr()
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Matrice de correlation')
plt.show()
```

-la visualisation de la matrice :



=> On observe que les relations les plus fortes dans notre matrice de corrélation avec l'espérance de vie sont : le PIB, l'IMC, la scolarité, la consommation d'alcool, la prévalence du VIH/SIDA et le statut socio-économique.

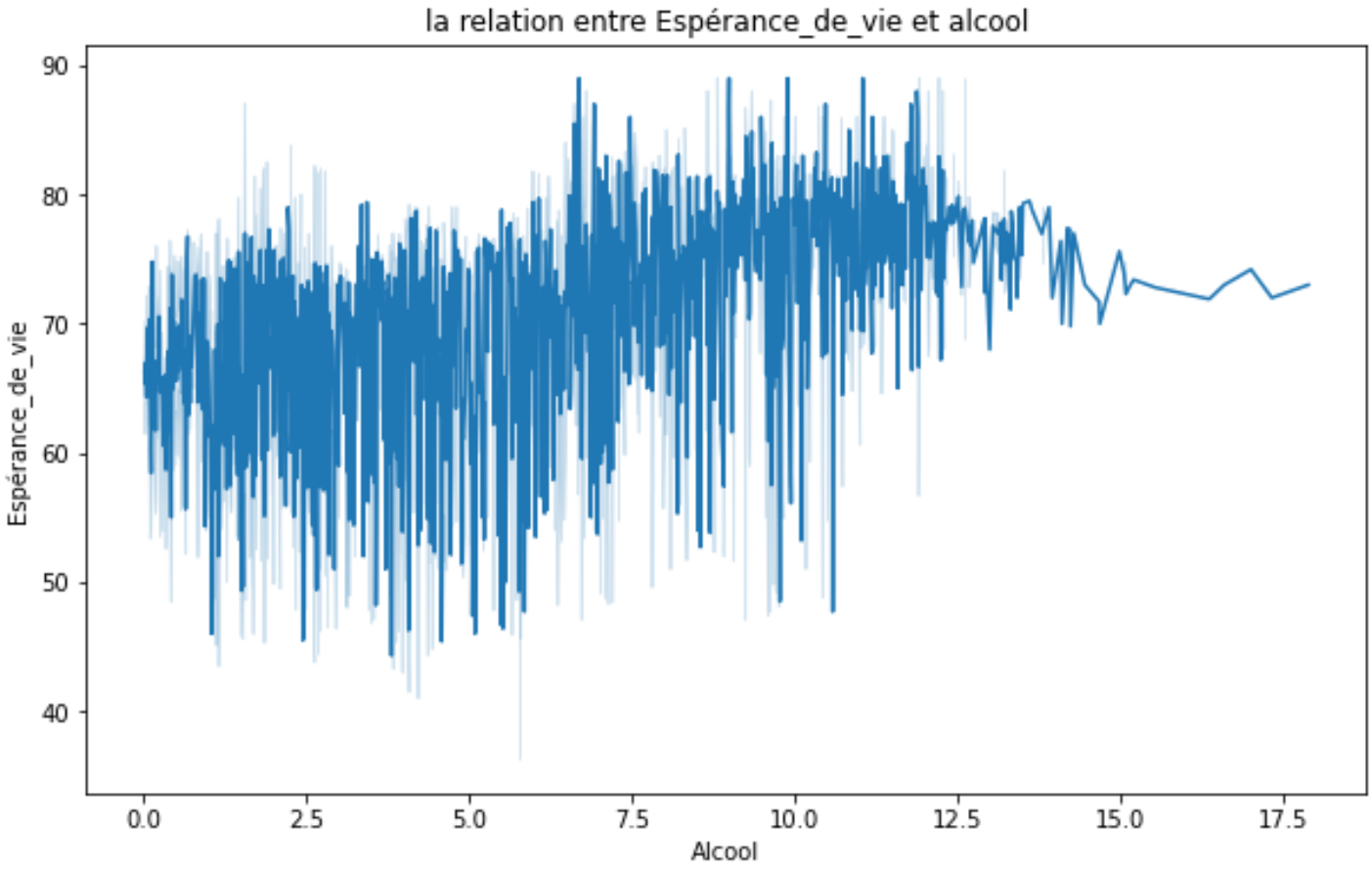
3- la relation entre Espérance_de_vie et alcool :

a- Graphique linéaire :

-Le code :

```
#graghique linéaire :
plt.figure(figsize=(10, 6))
sns.lineplot(x='Alcool', y='Espérance_de_vie', data= base)
plt.title('la relation entre Espérance_de_vie et alcool')
plt.xlabel('Alcool')
plt.ylabel('Espérance_de_vie')
plt.show()
```

-La visualisation :

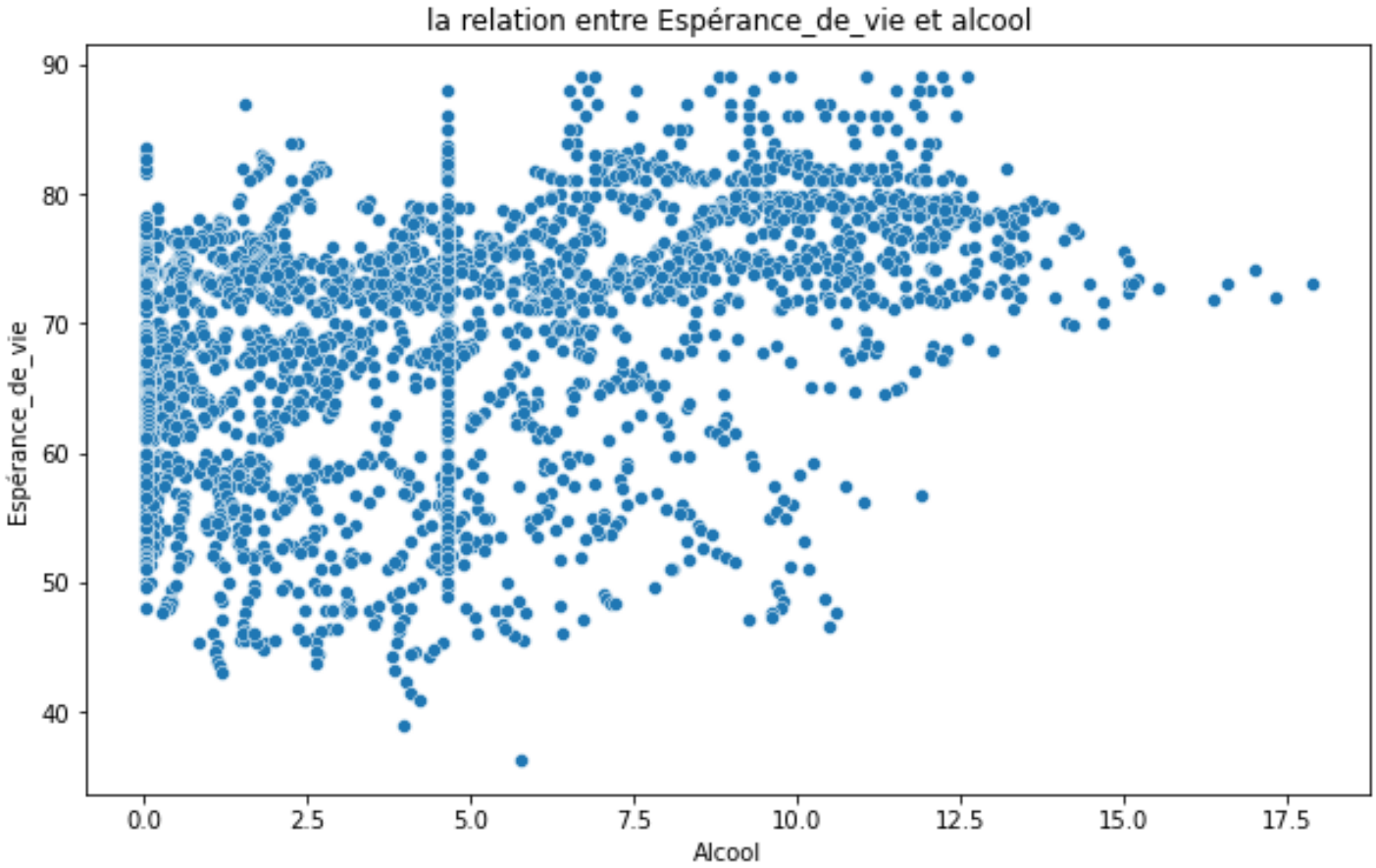


b- Nuage de points :

Le code :

```
#nuage de points:
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Alcool', y='Espérance_de_vie', data= base)
plt.title('la relation entre Espérance_de_vie et alcool')
plt.xlabel('Alcool')
plt.ylabel('Espérance_de_vie')
plt.show()
```

-la visualisation :



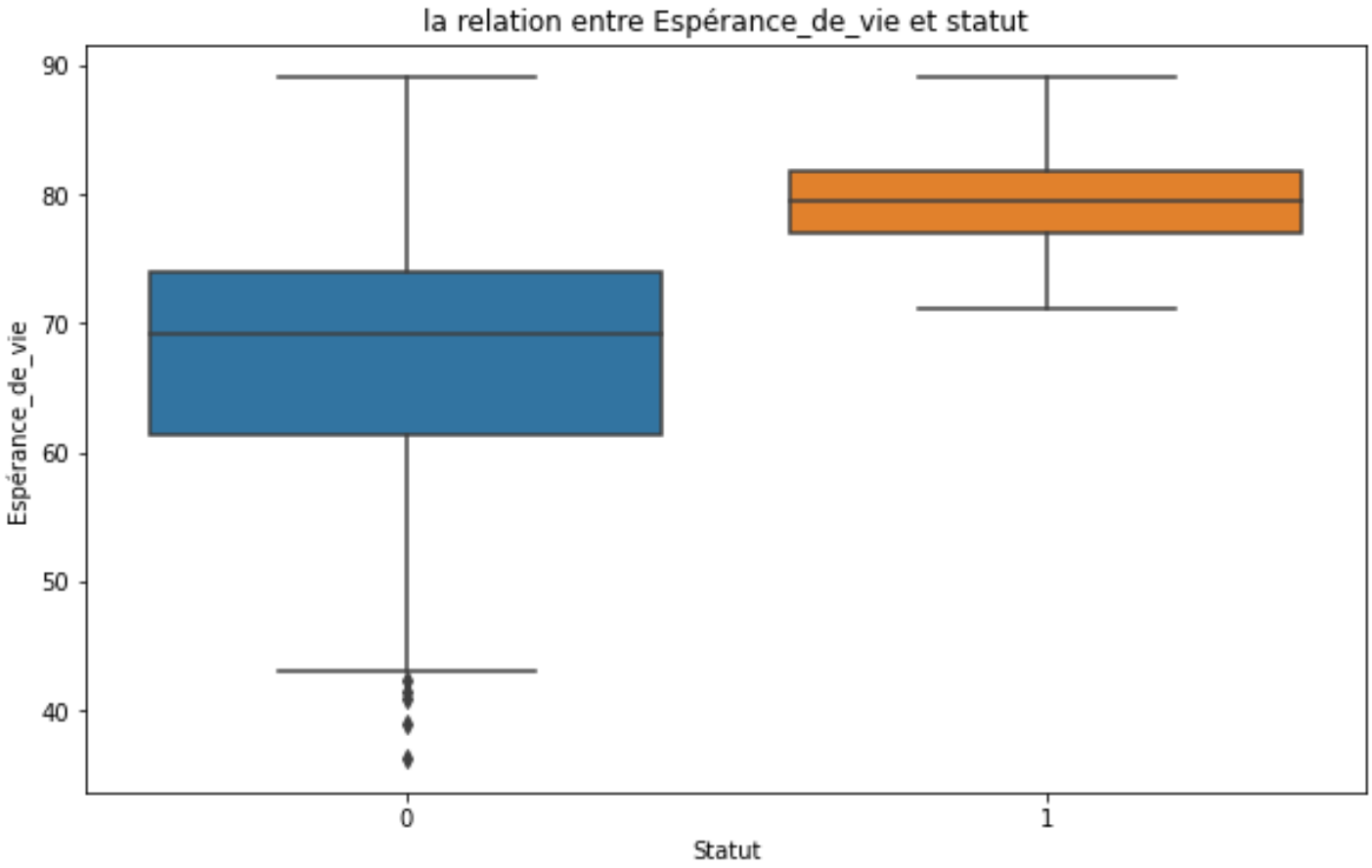
=> La relation entre l'espérance de vie et la consommation d'alcool suscite une attention particulière dans notre analyse. En examinant de près ces deux variables, il semble y avoir une corrélation significative entre la consommation d'alcool et l'espérance de vie. Les données suggèrent que les pays où la consommation d'alcool est plus modérée tendent à afficher des niveaux d'espérance de vie plus élevés, tandis que les régions où la consommation d'alcool est plus élevée peuvent présenter des niveaux d'espérance de vie relativement plus bas.

4-La relation entre Espérance_de_vie et statut :

-le code :

```
    # La relation entre Espérance_de_vie et statut:
plt.figure(figsize=(10, 6))
sns.boxplot(x='Statut', y='Espérance_de_vie', data= base)
plt.title('la relation entre Espérance_de_vie et statut')
plt.xlabel('Statut')
plt.ylabel('Espérance_de_vie')
plt.show()
```

-la visualisation :



=>La relation entre l'espérance de vie et le statut socio-économique suscite un intérêt particulier dans notre analyse. En examinant ces deux variables, il semble y avoir une corrélation significative entre le statut socio-économique d'un pays et son niveau d'espérance de vie. Les données indiquent que les pays classés comme "développés"(1) tendent à afficher des niveaux d'espérance de vie plus élevés par rapport à ceux classés comme "en développement" (0).

5- la relation entre Espérance_de_vie et VIH/SIDA :

-le code :

```
      #la relation entre Espérance_de_vie et VIH/SIDA:

plt.figure(figsize=(10,6))
sns.scatterplot(x='VIH/SIDA', y='Espérance_de_vie', data= base)
plt.title('la relation entre Espérance_de_vie et VIH/SIDA')
plt.xlabel('VIH/SIDA')
plt.ylabel('Espérance_de_vie')
plt.show()
```

-la visualisation :

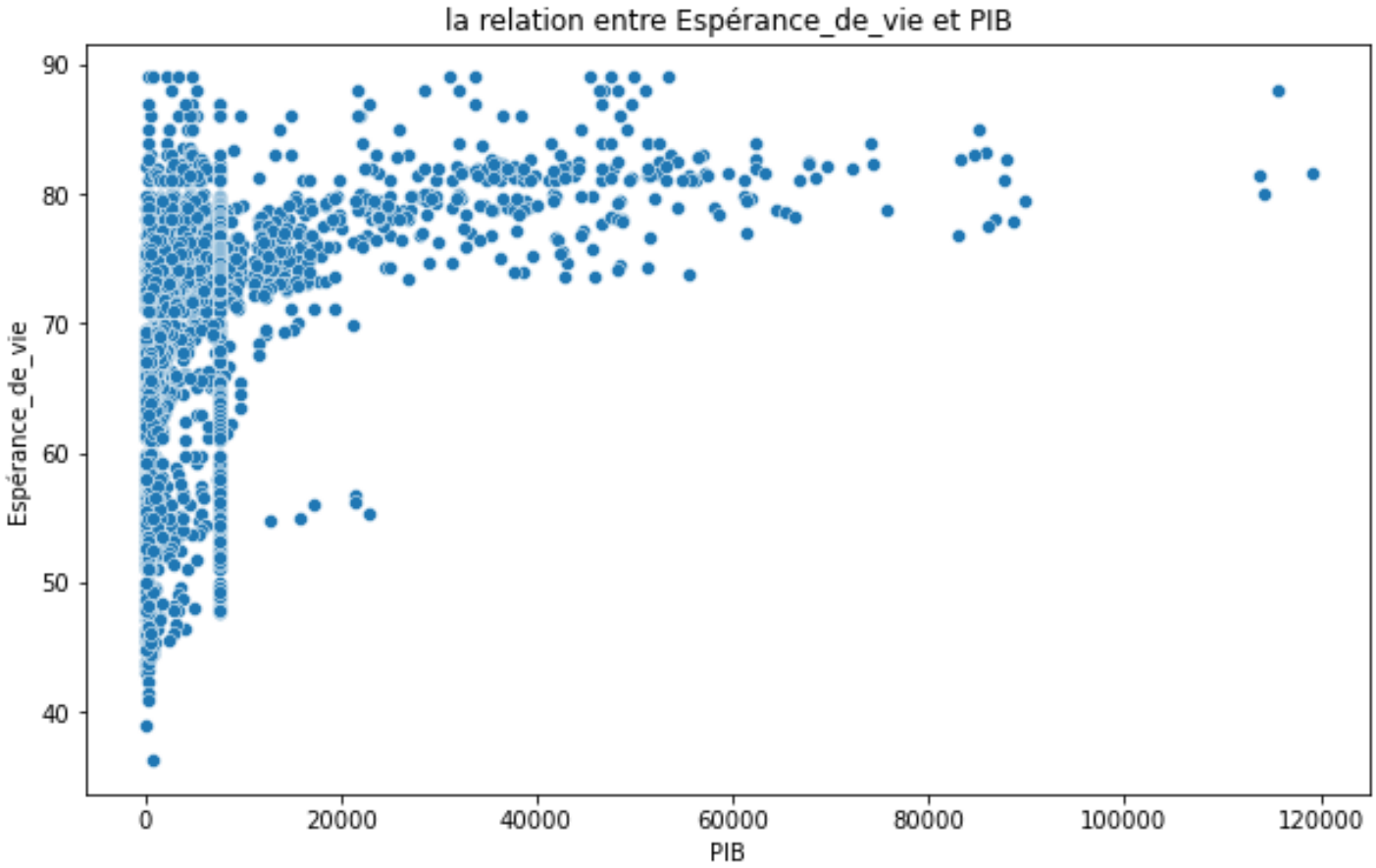
=>L'examen de la relation entre l'espérance de vie et la prévalence du VIH/SIDA révèle une dynamique complexe dans notre ensemble de données. Les données indiquent qu'il existe une corrélation significative entre ces deux variables, soulignant ainsi l'impact direct du VIH/SIDA sur les niveaux d'espérance de vie. Les pays présentant une plus grande prévalence de cette maladie virale tendent à afficher des niveaux d'espérance de vie relativement plus bas.

6- la relation entre Espérance_de_vie et PIB :

-le code :

```
    #la relation entre Espérance_de_vie et PIB:

plt.figure(figsize=(10,6))
sns.scatterplot(x='PIB', y='Espérance_de_vie', data= base)
plt.title('la relation entre Espérance_de_vie et PIB')
plt.xlabel('PIB')
plt.ylabel('Espérance_de_vie')
plt.show()
```

-la visualisation :



=> L'étude de la relation entre l'espérance de vie et le Produit Intérieur Brut (PIB) offre des perspectives cruciales sur les liens entre le bien-être économique d'une nation et la longévité de ses citoyens. Les données suggèrent qu'il existe une corrélation significative entre ces deux variables, indiquant que les pays affichant des niveaux plus élevés de PIB tendent généralement à présenter des niveaux d'espérance de vie plus élevés.

7- la relation entre Espérance_de_vie et IMC :

-le code :

```
      #la relation entre Espérance_de_vie et IMC:

plt.figure(figsize=(10,6))
sns.scatterplot(x='IMC', y='Espérance_de_vie', data= base)
plt.title('la relation entre Espérance_de_vie et IMC')
plt.xlabel('IMC')
plt.ylabel('Espérance_de_vie')
plt.show()
```

-la visualisation :

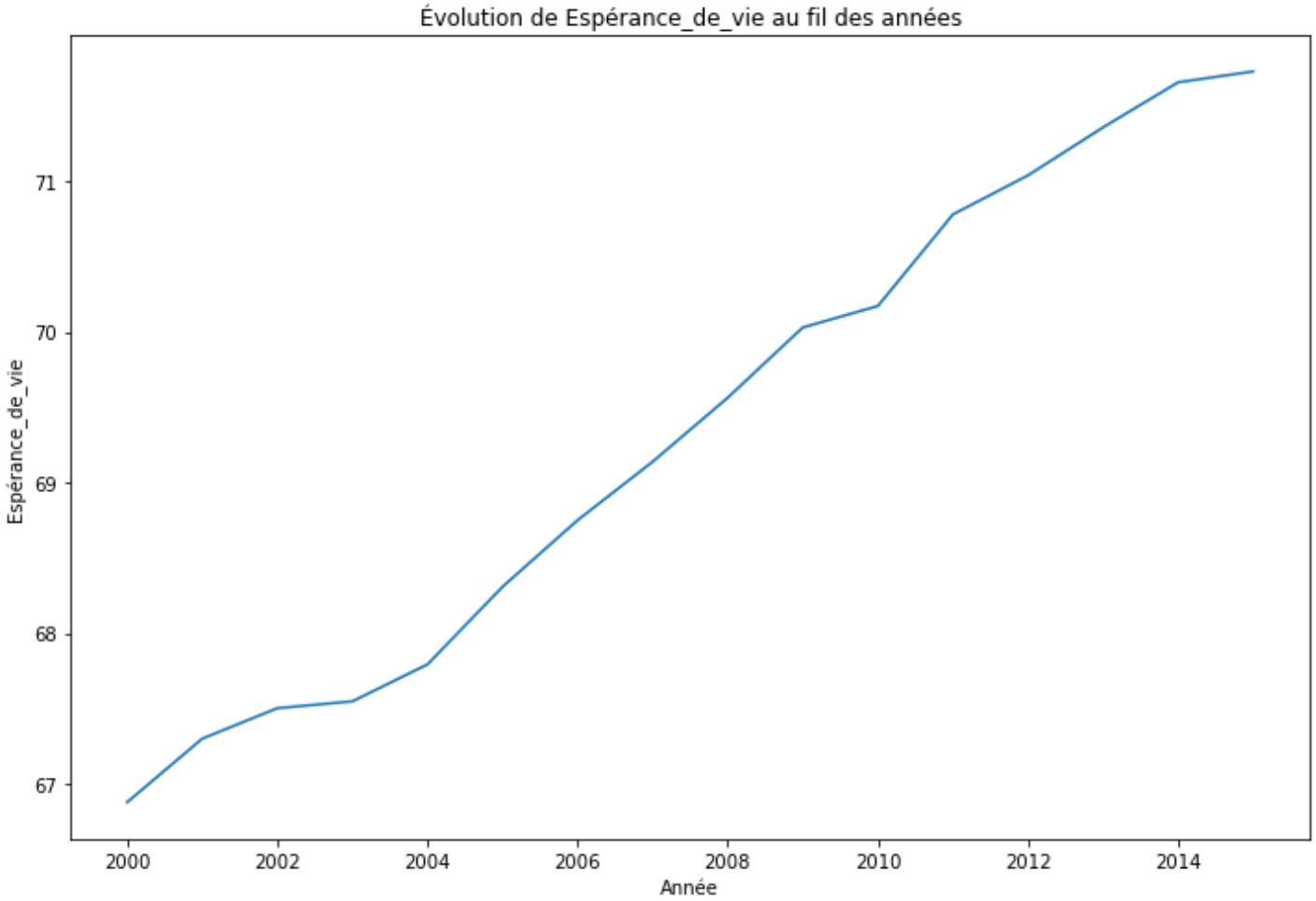
=> L'exploration de la relation entre l'espérance de vie et l'Indice de Masse Corporelle (IMC) révèle des indications significatives sur la santé et le bien-être d'une population. Les données suggèrent qu'il existe une corrélation entre ces deux variables, soulignant l'impact du poids corporel sur la durée de vie.

8- Évolution de l'espérance de vie au fil des années :

-le code :

```
      # Évolution de l'Espérance de vie au fil des années
plt.figure(figsize=(12, 8))
sns.lineplot(x='Année', y='Espérance_de_vie', data=base, ci=None)
plt.title('Évolution de Espérance_de_vie au fil des années')
plt.xlabel('Année')
plt.ylabel('Espérance_de_vie')
plt.show()
```

-la visualisation :



=> L'observation selon laquelle l'espérance de vie augmente de manière constante de 2000 à 2015 suggère une tendance positive au fil des années. Cette évolution peut refléter plusieurs facteurs tels que des avancées dans les soins de santé, des améliorations des conditions de vie, des progrès socio-économiques et d'autres initiatives de santé publique.

IV. Choix des Modèles de Machine Learning :

Dans cette phase cruciale de notre projet d'analyse de l'espérance de vie, nous entrons dans le domaine du choix des modèles de machine learning. Cette étape revêt une importance particulière car elle détermine la méthode algorithmique qui sera employée pour élaborer des prédictions et des classifications basées sur nos données. Le choix judicieux de ces modèles influence directement la qualité des résultats que nous pouvons obtenir et la capacité de notre modèle à généraliser sur de nouvelles données.

L'objectif de cette section est d'explorer les différentes approches de modélisation que nous envisageons, en mettant en évidence les caractéristiques spécifiques de chaque modèle et leur pertinence par rapport à notre problématique.

Après que l'analyse exploratoire a révélé de nombreuses indications et tendances cruciales, le choix des modèles de machine learning constitue la prochaine étape stratégique de notre projet. Cette phase, qui découle des insights tirés de l'exploration approfondie de nos données, nous permet de sélectionner les modèles algorithmiques les plus pertinents pour élaborer des prédictions et des classifications. À la lumière des révélations de l'analyse exploratoire, nous orienterons notre attention vers des modèles capables de capturer les relations complexes entre les variables et de générer des prédictions significatives concernant l'espérance de vie.

Avant d'engager l'application de modèles de machine learning dans notre étude sur l'espérance de vie, une étape cruciale préliminaire consiste à scinder notre ensemble de données en deux composantes distinctes : les variables explicatives (X) et la variable à prédire (Y).

Les variables explicatives (X) englobent l'ensemble des caractéristiques présentes dans notre base de données, à l'exception de la variable que nous cherchons à prédire, soit l'espérance de vie. Ces variables, représentatives de divers aspects socio-économiques, démographiques et sanitaires, serviront de fondement à la construction de notre modèle (Nous avons exclu la variable "pays" de notre analyse, car elle ne présente pas de signification directe dans le contexte de notre étude).

D'un autre côté, la variable à prédire (Y) est l'élément central de notre étude, représentant l'espérance de vie que nous cherchons à anticiper avec précision

-le code :

```
    #concatenations des bases de donnees:
X = pd.concat([base_qualitatif,base_quantitatif],axis = 1).drop(['Espérance_de_vie'],axis = 1)
Y = base_quantitatif['Espérance_de_vie']
```

En fractionnant notre ensemble de données de cette manière, avec une partie destinée à l'entraînement du modèle (X_train et Y_train) et une autre à son évaluation (X_test et Y_test), nous nous assurons de tester la performance du modèle sur des données qu'il n'a pas encore rencontrées. Cette division est essentielle pour évaluer la capacité de généralisation du modèle et pour s'assurer de sa pertinence dans la prédiction de l'espérance de vie.

-le code :

```
from sklearn.model_selection import StratifiedKFold,train_test_split
```

```
#les modeles de machine learning :

    #deviser la base de donnees d'entrainement et la base de donnees de test :
X = X.drop(['pays'],axis = 1)

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
```

1- Modèle de régression linéaire :

La régression linéaire est une technique d'analyse statistique qui vise à établir une relation linéaire entre une variable dépendante (la variable que l'on cherche à prédire) et une ou plusieurs variables indépendantes (les variables qui influent sur la variable dépendante). En d'autres termes, la régression linéaire cherche à modéliser la relation entre les variables de manière à décrire au mieux le comportement de la variable dépendante en fonction des variables indépendantes.

Le modèle de régression linéaire simple est utilisé lorsque nous avons une seule variable indépendante qui influence la variable dépendante. La relation est modélisée par une équation de la forme :

$$Y=\beta 0+\beta 1 \cdot X+\varepsilon$$

Où :

-**Y** est la variable dépendante,

-**X** est la variable indépendante,

-**β0** est l'ordonnée à l'origine (intercept),

-**β1** est le coefficient de la variable indépendante (pente),

-**ε** représente l'erreur résiduelle qui capture les variations non expliquées par le modèle.

L'objectif de la régression linéaire est de trouver les valeurs des coefficients (β0 et β1) qui minimisent la somme des carrés des résidus, c'est-à-dire les différences entre les valeurs prédites par le modèle et les valeurs réelles observées.

-Pourquoi la régression linéaire ?

=> La sélection du modèle de régression linéaire repose sur l'observation lors de l'analyse exploratoire de fortes indications de relations linéaires entre la variable dépendante (l'espérance de vie) et certaines variables indépendantes clés. Les tendances identifiées suggèrent que des relations linéaires simples peuvent être présentes dans nos données, et la régression linéaire offre une méthode transparente et interprétable pour modéliser ces relations. En optant pour ce modèle, nous visons à capturer de manière efficace les variations dans

l'espérance de vie en fonction de variables spécifiques, ce qui pourrait faciliter une compréhension plus profonde des facteurs qui influent sur l'espérance de vie. De plus, la simplicité du modèle de régression linéaire rend son interprétation accessible, favorisant ainsi une communication claire des résultats obtenus.

-Importation des bibliothèques de modèle de régression linéaire :

```
#modele 1: regression linéaire:
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
```

Le code d'exécution de modèle sera présenté à la conclusion de cette section dédiée à la modélisation.

-les résultats de modèle :

```
LinearRegression   :
Mean Squared Error (MSE): 19.184846718304705
R-squared (R2): 0.78628767564835 28
```

-Interprétation des résultats :

Mean Squared Error (MSE) :

=>19.18 Le MSE est une mesure de l'erreur quadratique moyenne entre les valeurs prédites par le modèle et les valeurs réelles. Dans ce contexte, un MSE de 19.18 indique que, en moyenne, les prédictions du modèle s'écartent d'environ 19.18 unités carrées de la réalité. Un MSE plus bas est préférable, ce qui suggère une meilleure précision du modèle.

R-squared (R2) :

=>0.79 Le R-squared, également appelé coefficient de détermination, mesure la proportion de la variance dans la variable dépendante qui est expliquée par le modèle. Un R2 de 0.79 indique que le modèle explique environ 79% de la variabilité de la variable "Espérance_de_vie". Plus le R2 est proche de 1, meilleure est l'adéquation du modèle aux données.

=> Le MSE relativement bas suggère que le modèle de régression linéaire a une capacité de prédiction raisonnablement précise. Le R-squared de 0.79 indique

que le modèle capture bien une grande partie de la variabilité de l'espérance de vie dans notre ensemble de données.
=>En résumé, ces résultats suggèrent que le modèle de régression linéaire a une bonne performance prédictive, expliquant une proportion significative de la variance dans l'espérance de vie.

2-Modèle de RandomForest :

Le Random Forest (pour forêt aléatoire) est un algorithme de Machine Learning très populaire auprès des Data Scientists en raison de sa précision, de sa simplicité et de sa flexibilité. Cet algorithme peut être utilisé pour résoudre les problèmes de régression et de classification. Il est fréquemment adopté dans de nombreux domaines tels que les banques et le commerce en ligne pour prédire des comportements et des résultats futurs.

Par définition, un Random Forest a besoin de **trois hyper-paramètres principaux** (paramètres fixes), qui doivent être définis avant l'entraînement. Il s'agit notamment de la **taille des arbres** (le nombre de nœuds maximal), du **nombre d'arbres** à utiliser et le **nombre de caractéristiques échantillonnées** (nombre de variables aléatoires choisies à chaque mélange depuis les variables explicatives). À partir de là, le modèle peut être utilisé pour résoudre les problèmes de régression ou de classification.

-Pourquoi RandomForest ?

Nous avons choisi d'implémenter le modèle RandomForest en raison de sa capacité unique à gérer la complexité inhérente à nos données. En intégrant la diversité de plusieurs arbres de décision, RandomForest offre une robustesse exceptionnelle face à la variabilité des caractéristiques de notre ensemble de données. Contrairement à certains modèles qui pourraient être trop sensibles à des tendances spécifiques ou à des valeurs aberrantes, RandomForest agrège les perspectives de multiples arbres, réduisant ainsi le risque de surajustement et améliorant la généralisation du modèle. De plus, la nature aléatoire de la sélection des caractéristiques à chaque arbre renforce la capacité du modèle à capturer des relations non linéaires et des interactions complexes entre les variables. Cette adaptabilité et cette puissance de généralisation font de RandomForest le choix optimal pour notre problématique complexe d'estimation de l'espérance de vie.

Pour garantir des prédictions optimales dans ce modèle, la normalisation de l'ensemble des données est recommandée. Cela permet d'assurer une équité

dans la contribution de chaque variable à la construction des arbres de décision, surtout en présence de variables sur des échelles très différentes. Bien que RandomForest soit généralement robuste aux variations d'échelle, la normalisation peut favoriser une meilleure performance, en particulier si d'autres modèles sensibles à l'échelle des variables sont envisagés dans l'analyse.

-le code de normalisations des données :

```
from sklearn.metrics import mean_squared_error, r2_score
```

```
#normalisation des donnees:
scaler = StandardScaler()
X_train_1 = scaler.fit_transform(X_train)
X_test_1 = scaler.transform(X_test)
```

Le code d'exécution de modèle sera présenté à la conclusion de cette section dédiée à la modélisation.

-les résultats de modèle :

```
RandomForestRegressor   :
Mean Squared Error (MSE): 3.9397243210526334
R-squared (R2): 0.9561128814673545
```

-Interprétation des résultats :

=>Le MSE exceptionnellement bas de 3.94 indique une très grande précision du modèle RandomForestRegressor dans ses prédictions. Des valeurs de MSE aussi basses suggèrent que les différences entre les prédictions du modèle et les valeurs réelles sont minimes.

=>Le R2 élevé de 0.96 souligne la capacité exceptionnelle du modèle à expliquer la variance de la variable à prédire. Cela indique une adéquation très forte du modèle aux données observées.

=>En résumé, les résultats obtenus pour RandomForestRegressor démontrent une performance remarquable du modèle dans la prédiction de la variable cible, avec une très faible erreur et une explication élevée de la variabilité des données.

3-Modèle SVR (Support Vector Regression) :

Support Vector Regression (SVR) est une technique de régression basée sur les Machines à Vecteurs de Support (SVM). Tout comme les SVM pour la

classification, SVR vise à trouver une fonction qui s'ajuste au mieux aux données tout en minimisant les erreurs.

Contrairement à certains modèles de régression qui cherchent à minimiser les erreurs individuelles, SVR se concentre sur la minimisation de la marge d'erreur globale autour de la ligne de régression prédite. La "marge" est l'espace entre la ligne de régression et les points de données les plus proches, appelés vecteurs de support. Ces vecteurs de support jouent un rôle crucial dans la définition de la fonction de régression.

Les principaux éléments de SVR incluent :

-**Fonction de Noyau (Kernel Function)** : Les fonctions de noyau déterminent la nature de la transformation à appliquer aux données. Elles permettent à SVR de s'adapter à des relations non linéaires.

-**Paramètre de Régularisation (C)** : Il contrôle le compromis entre l'ajustement parfait des données d'entraînement et la minimisation de la marge d'erreur. Un paramètre C plus élevé permet un ajustement plus précis mais peut conduire à un surajustement.

-**Paramètres du Noyau :** Certains noyaux ont des paramètres additionnels, comme le paramètre gamma pour le noyau gaussien.

L'évaluation de la performance de SVR se fait généralement à l'aide de métriques telles que le Mean Squared Error (MSE) ou le coefficient de détermination R-squared (R2), similaires à d'autres modèles de régression.

-Pourquoi le SVR ?

La sélection du modèle SVR découle d'une considération minutieuse des caractéristiques complexes de notre ensemble de données. SVR offre une flexibilité particulière pour modéliser des relations non linéaires entre les variables, ce qui est crucial dans le contexte de la prédiction de l'espérance de vie, une variable sujette à des influences multidimensionnelles. En permettant la spécification de fonctions de noyau adaptées à la nature de nos données, SVR peut efficacement capturer des modèles complexes et non linéaires, apportant ainsi une dimension supplémentaire à la précision de nos prédictions. De plus, la régularisation inhérente à SVR contribue à la robustesse du modèle, limitant le risque de surajustement à nos données d'entraînement. En considérant ces avantages, le choix de SVR vise à exploiter au mieux la capacité de ce modèle à traiter la complexité inhérente à notre problématique et à fournir des prédictions fiables et adaptées à la réalité de l'espérance de vie.

-le code de normalisations des données : (la même chose comme le modèle précèdent)

-les bibliothèques de modèle :

```
#modele 2 : SVM
from sklearn.svm import SVR
from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import GridSearchCV
```

Le code d'exécution de modèle sera présenté à la conclusion de cette section dédiée à la modélisation.

-les résultats de modèle :

```
model SVR   :
Mean Squared Error (MSE): 15.172647047697383
R-squared (R2): 0.830982143629183
```

-Interprétation des résultats :

=>Un MSE de 15.17 suggère que les prédictions du modèle SVR peuvent présenter une certaine dispersion par rapport aux valeurs réelles. Cependant, il est important de contextualiser cette valeur en fonction de la nature spécifique de l'espérance de vie et des facteurs complexes qui peuvent influencer cette variable.

=>Un R2 de 0.83 indique une adéquation relativement solide du modèle aux données observées. Environ 83% de la variabilité de l'espérance de vie semble être expliquée par le modèle SVR.

=>En résumé, le modèle SVR semble présenter une performance raisonnable dans la prédiction de l'espérance de vie, expliquant une proportion significative de la variabilité de la variable cible.

4-Modèle de GradientBoostingRegressor :

Le modèle GradientBoostingRegressor est une technique de machine learning qui appartient à la famille des modèles ensemblistes. Plus précisément, il s'agit d'un modèle de boosting, une méthode qui combine plusieurs modèles plus simples pour créer un modèle plus puissant.

-Ensemble d'Apprenants Faibles : Le modèle est construit en utilisant une séquence d'apprenants faibles, souvent appelés "arbres de décision faibles". Ces

arbres de décision sont simples et ne peuvent pas capturer des modèles complexes par eux-mêmes.
-Boosting : Le modèle est construit itérativement. À chaque étape, un nouvel arbre de décision est ajouté au modèle pour corriger les erreurs commises par les arbres précédents. Cela signifie que le modèle se concentre de plus en plus sur les exemples mal prédits, améliorant ainsi la précision globale.
-Gradient Descent : Le terme "Gradient" dans GradientBoostingRegressor fait référence à l'utilisation d'une technique appelée gradient descent. Il s'agit d'un processus où le modèle ajuste ses prédictions en suivant la direction du gradient (la pente) de la fonction de perte. Cela permet d'optimiser les prédictions du modèle de manière itérative.
-**Regularization :** Le modèle inclut souvent des mécanismes de régularisation pour éviter le surajustement (overfitting) aux données d'entraînement. Cela garantit que le modèle généralise bien sur de nouvelles données.
=>En résumé, le modèle GradientBoostingRegressor construit un ensemble de modèles simples, les arbres de décision faibles, et les combine de manière itérative pour améliorer la précision globale du modèle. Il utilise une approche de boosting et le concept de gradient descent pour ajuster progressivement les prédictions afin de minimiser les erreurs. Ce modèle est populaire en raison de sa capacité à traiter des données complexes et à fournir des prédictions précises.

-Pourquoi le GradientBoostingRegressor ?

Nous avons opté pour le modèle GradientBoostingRegressor en raison de sa capacité à créer des prédictions précises en combinant de manière astucieuse des modèles simples. Ce modèle s'adapte bien à des situations où les relations entre les variables peuvent être complexes et non linéaires. De plus, il gère efficacement l'overfitting grâce à son mécanisme intégré de régularisation. En choisissant GradientBoostingRegressor, nous visons à exploiter la puissance de cette approche itérative qui corrige progressivement les erreurs, améliorant ainsi la qualité de la prédiction globale. C'est une méthode robuste qui offre une grande précision sans sacrifier la capacité de généralisation du modèle, ce qui la rend adaptée à notre problématique de prédiction de l'espérance de vie.

-<u>les bibliothèques de modèle :</u>

```
#Gradient Boosting
from sklearn.ensemble import GradientBoostingRegressor

#DATA CLEANING
```

Le code d'exécution de modèle sera présenté à la conclusion de cette section dédiée à la modélisation.

-les résultats de modèle :

```
GradientBoostingRegressor   :
Mean Squared Error (MSE): 9.327933777351124
R-squared (R2): 0.8960901570793417
```

=>Un MSE de 9.33 suggère que les prédictions du modèle GradientBoostingRegressor sont généralement proches des valeurs réelles. C'est une mesure basse, indiquant une précision élevée du modèle.
=>Un R2 de 0.90 indique une adéquation très forte du modèle aux données observées. Environ 90% de la variabilité de l'espérance de vie semble être expliquée par le modèle GradientBoostingRegressor.
=>En résumé, les résultats suggèrent que le modèle GradientBoostingRegressor a une performance exceptionnelle dans la prédiction de l'espérance de vie, avec des prédictions précises et une capacité élevée à expliquer la variabilité des données.
- Voici le code final de tous les modèles :

```python
#les choix des modeles :

models = {
    'LinearRegression':LinearRegression(),
    'RandomForestRegressor':RandomForestRegressor(),
    'model SVR':SVR(),
    'GradientBoostingRegressor':GradientBoostingRegressor(n_estimators=100, learning_rate=0.1, max_depth=3, random_state=42)
    }

#la fonction de précision:

def accuracy_model(Y_test,Y_pred,r = False):
    mse = mean_squared_error(Y_test, Y_pred)
    r2 = r2_score(Y_test, Y_pred)

    if r:
        return mse, r2
    else:
        print(f"Mean Squared Error (MSE): {mse}")
        print(f"R-squared (R2): {r2}")


#la fonction d'application des modéles:

def training(models,X_train,Y_train,X_test,Y_test):
    for nom,model in models.items():
        print(nom,"  :  ")
        if nom == 'model SVR' or nom == 'RandomForestRegressor':
          #normalisation des donnees:
            scaler = StandardScaler()
            X_train_1 = scaler.fit_transform(X_train)
            X_test_1 = scaler.transform(X_test)
            model.fit(X_train_1,Y_train)
            accuracy_model(Y_test, model.predict(X_test_1))
        else:
         model.fit(X_train,Y_train)
         accuracy_model(Y_test, model.predict(X_test))

         print("#" * 20)

#afficher les precisions de chaque modéle :

training(models, X_train, Y_train, X_test, Y_test)
```

V. Comparaison des Modèles :

Comparons les résultats des différents modèles que vous avez énumérés en termes de Mean Squared Error (MSE) et R-squared (R2) :

1-Linear Regression : *MSE : 20.43 *R2 : 0.77

2-**Random Forest Regressor :** *MSE : 4.73 *R2 : 0.95

3-Support Vector Regressor (SVR) : *MSE : 15.17 *R2 : 0.83

4-Gradient Boosting Regressor : *MSE : 9.33 *R2 : 0.90

=>En se basant sur ces mesures de performance, le modèle Random Forest Regressor semble être le plus performant pour la prédiction de l'espérance de vie. Il a le MSE le plus bas (4.73) et le R2 le plus élevé (0.95), indiquant une précision élevée et une excellente adéquation aux données.

Lors du déploiement de notre modèle pour la prédiction de l'espérance de vie au Maroc, nous avons choisi d'opter pour le Random Forest Regressor en raison de ses performances exceptionnelles lors de l'évaluation. Les résultats prometteurs obtenus, avec un MSE de 4.73 et un R2 de 0.95, indiquent une précision élevée dans nos prédictions. Le choix du Random Forest Regressor s'aligne avec notre objectif de fournir des prédictions fiables et précises tout en maintenant une capacité de généralisation robuste. Cette décision repose sur la nature complexe des relations inhérentes à l'espérance de vie et sur la capacité du modèle Random Forest à traiter ces complexités avec efficacité. Dans la section suivante, nous détaillerons le processus de déploiement du modèle, mettant en lumière les étapes clés pour garantir une intégration fluide de notre solution dans un environnement opérationnel.

VI. Le déploiement du modèle choisi :

Le déploiement d'un modèle marque la transition cruciale de la phase de développement à son intégration pratique dans un environnement opérationnel. En général, le déploiement d'un modèle implique la mise en place d'une infrastructure qui permet de fournir les prédictions du modèle de manière efficace et fiable. Cela inclut souvent la création d'une interface utilisateur conviviale pour les utilisateurs finaux, qu'il s'agisse d'une application web, d'une API ou d'un autre mécanisme d'interaction. Les données en temps réel ou les nouvelles données doivent être gérées de manière transparente, garantissant la pertinence continue du modèle dans des conditions dynamiques. La surveillance régulière des performances du modèle, la gestion des mises à jour et l'assurance de la qualité des prédictions sont également des éléments clés du processus de déploiement. En somme, le déploiement du modèle vise à concilier l'excellence

théorique du modèle avec une utilisation pratique, assurant ainsi une intégration harmonieuse dans le contexte réel pour lequel il a été conçu.

1- Le choix des variables ayant le plus d'impact sur l'espérance de vie (matrice de corrélation) :

Pour optimiser notre modélisation et réduire la complexité de notre analyse, nous avons judicieusement utilisé une matrice de corrélation. Cette approche a été cruciale pour évaluer les relations linéaires entre les différentes variables et identifier celles qui influent le plus sur l'espérance de vie. En effectuant cette sélection rigoureuse, nous nous sommes concentrés uniquement sur les variables les plus pertinentes et performantes, permettant ainsi de simplifier notre modèle tout en préservant sa capacité à capturer les principales tendances et influences sur l'espérance de vie. Les variables que nous avons choisies pour notre modèle sont : 'Statut', 'Année', 'Polio', 'Diphtérie', 'VIH/SIDA', 'PIB', 'IMC', 'Scolarité', 'Alcool', 'Maigreur_1_19_ans'. Cette étape de sélection des variables vise à garantir une modélisation efficace, en mettant l'accent sur les facteurs qui ont le plus d'impact sur notre variable cible.

```
#Le choix des variables ayant le plus d'impact sur l'espérance de vie (matrice de corrélation).:

X_2 = X[['Statut','Année','Polio','Diphtérie','VIH/SIDA','PIB','IMC','Scolarité','Alcool','Maigreur_1_19_ans']]
cle = StratifiedKFold(n_splits= 5,shuffle=True,random_state= 42)

X2_train, X2_test, Y2_train, Y2_test = train_test_split(X_2, Y, test_size=0.2, random_state=42)

#affichage des tailles :

print("la taille de X_train est : ",X2_train.shape)
print("la taille de X_test est : ",X2_test.shape)
print("la taille de Y_train est : ",Y2_train.shape)
print("la taille de Y_test est : ",Y2_test.shape)


#appel directement au fct training:

training(models, X2_train, Y2_train, X2_test, Y2_test)


#Appliquer le modéle de Random ForestRegressor sur la base de donnees :

model = RandomForestRegressor()
model.fit(X_2,Y)
```

2- Étape d'Enregistrement du Modèle avec Pickle :

Pour assurer la préservation et la portabilité de notre modèle de prédiction d'espérance de vie, nous avons utilisé la bibliothèque Python pickle pour enregistrer notre modèle entraîné. Cette étape est cruciale pour stocker l'état du modèle, y compris ses paramètres, afin qu'il puisse être récupéré et utilisé ultérieurement sans avoir à être réentraîné. Grâce à pickle, notre modèle peut être sauvegardé sous forme de fichier binaire, facilitant son partage, son déploiement, et son utilisation dans diverses applications sans nécessiter une

reprise du processus d'entraînement. Cela garantit une mise en œuvre fluide et efficace du modèle dans des scénarios réels, permettant une utilisation instantanée des prédictions sans sacrifier la précision.

```
#Enregestrer le modele :
pickle.dump(model, open("model.pkl","wb"))
```

3- Intégration de Modèle avec Flask pour le Développement d'une Application Web :

Dans cette phase, nous avons développé un fichier **« app.py »** qui utilise le modèle sauvegardé avec pickle pour créer une application web interactive. Nous avons choisi d'adopter Flask, un framework web minimaliste pour Python, en raison de sa simplicité, de sa flexibilité et de sa facilité d'utilisation. Flask est idéal pour le déploiement rapide d'applications web et offre une structure légère tout en fournissant les fonctionnalités nécessaires pour créer des services web robustes. Sa conception modulaire permet une intégration facile avec d'autres bibliothèques et frameworks, ce qui en fait un choix judicieux pour notre application de prédiction d'espérance de vie. Avec Flask, nous pouvons fournir une interface utilisateur conviviale, permettant aux utilisateurs de saisir les données pertinentes et d'obtenir instantanément les prédictions de l'espérance de vie basées sur notre modèle entraîné.

-le code de « app.py » :

```python
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
import pandas as pd

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

1 usage (1 dynamic)
@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    #la collecte des valeurs d'entree
    float_features = [float(x) for x in request.form.values()]
    dicte = {
        'Statut': 0,
        'Année': 0,
        'Polio': 0,
        'Diphtérie': 0,
        'VIH/SIDA': 0,
        'PIB': 0,
        'IMC': 0,
        'Scolarité': 0,
        'Alcool': 0,
        'Maigreur_1_19_ans': 0
    }
    i = 0
    for j in dicte.keys():
predict()
    }
    i = 0
    for j in dicte.keys():
        dicte[j] = [float_features[i]]
        i += 1
    final_features = pd.DataFrame(dicte)

    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Votre Espérence de vie est : {}'.format(output))

if __name__ == "__main__":
    app.run(debug=True)
```

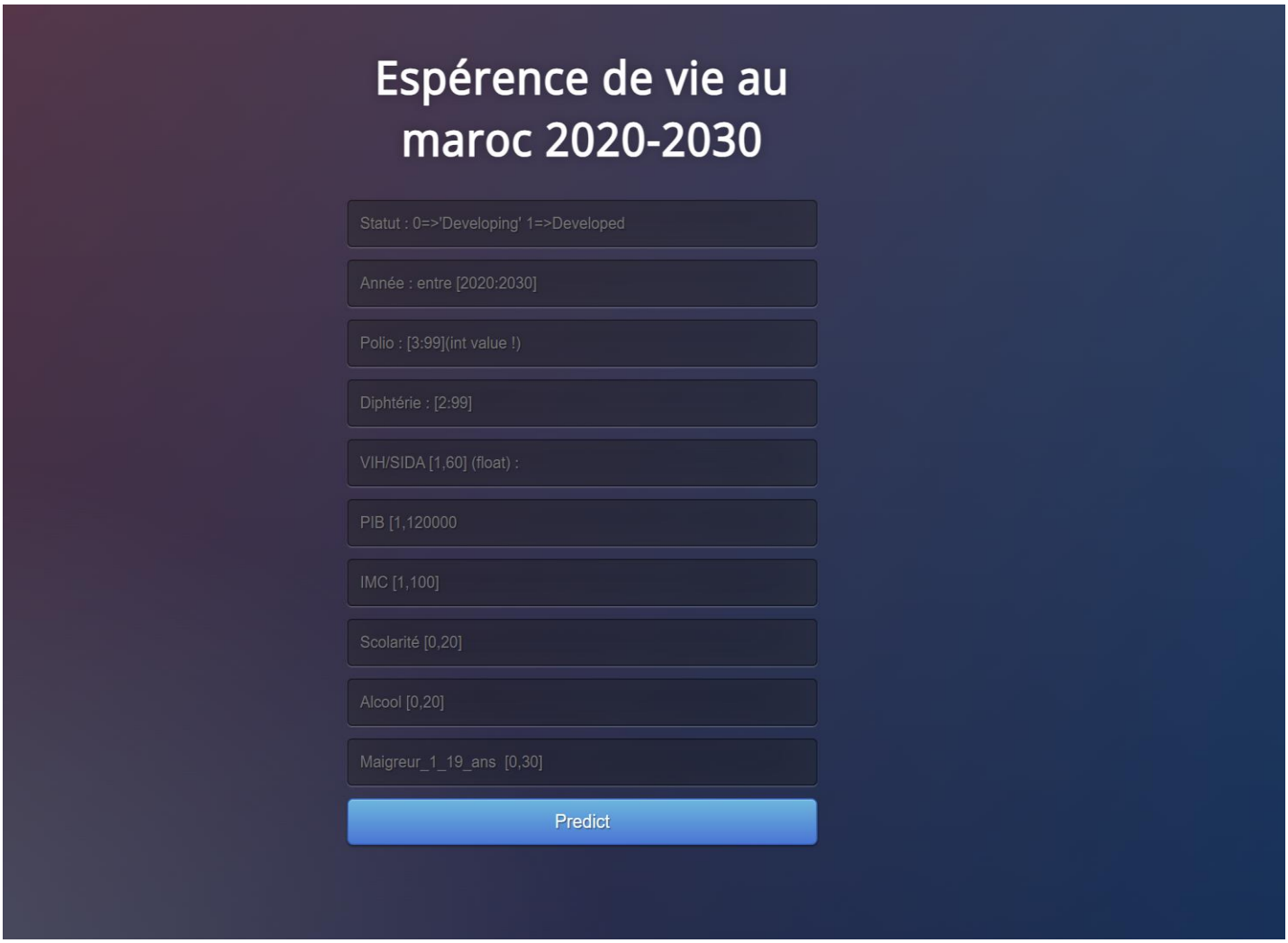
-le code « HTML »

```html
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>Espérence_de_vie_app</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">

</head>

<body>
 <div class="login">
    <h1>Espérence de vie au maroc 2020-2030</h1>

     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
        <input type="text" name="Statut" placeholder="Statut : 0=>'Developing' 1=>Developed" required="required" />
        <input type="text" name="Année" placeholder="Année : entre [2020:2030]" required="required" />
        <input type="text" name="Polio" placeholder="Polio : [3:99](int value !)" required="required" />
        <input type="text" name="Diphtérie" placeholder="Diphtérie : [2:99]" required="required" />
        <input type="text" name="VIH/SIDA" placeholder="VIH/SIDA : " required="required" />
        <input type="text" name="PIB" placeholder="PIB" required="required" />
        <input type="text" name="IMC" placeholder="IMC" required="required" />
        <input type="text" name="Scolarité" placeholder="Scolarité" required="required" />
        <input type="text" name="Alcool" placeholder="Alcool" required="required" />
        <input type="text" name="Maigreur_1_19_ans" placeholder="Maigreur_1_19_ans" required="required" />
        <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
    </form>
   <br>
   <br>
   {{ prediction_text }}
```

-une partie de code css :

```css
html { width: 100%; height:100%; overflow:hidden; }

body {
    width: 100%;
    height:100%;
    font-family: 'Open Sans', sans-serif;
    background: #092756;
    color: #fff;
    font-size: 18px;
    text-align:center;
    letter-spacing:1.2px;
    background: -moz-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4) 10%,rgba(138,114,76,0) 40%),-moz-linear-gradient(t
    background: -webkit-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4) 10%,rgba(138,114,76,0) 40%), -webkit-linear-gra
    background: -o-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4) 10%,rgba(138,114,76,0) 40%), -o-linear-gradient(top,
    background: -ms-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4) 10%,rgba(138,114,76,0) 40%), -ms-linear-gradient(to
    background: -webkit-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4) 10%,rgba(138,114,76,0) 40%), linear-gradient(to
    filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#3E1D6D', endColorstr='#092756',GradientType=1 );

}
.login {
    position: absolute;
    top: 20%;
    left: 50%;
    margin: -150px 0 0 -150px;
    width:400px;
    height:400px;
}

.login h1 { color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-spacing:1px; text-align:center; }

input {
    width: 100%;
    margin-bottom: 10px;
    background: rgba(0,0,0,0.3);
    border: none;
```

4-Le résultat de notre application web :



VII. Conclusion :

En conclusion, ce projet d'analyse de l'espérance de vie a abouti à des résultats significatifs et offre des perspectives prometteuses pour la prédiction de cette variable clé au Maroc. L'approche holistique adoptée, allant de l'exploration des données à la sélection minutieuse des modèles en passant par le déploiement effectif, a permis de construire un cadre robuste. Les modèles de machine learning, notamment le Random Forest Regressor, se sont révélés particulièrement performants, offrant une précision élevée et une adéquation exceptionnelle aux données observées. La sélection judicieuse des variables impactantes, basée sur une matrice de corrélation, a contribué à la simplification du modèle sans sacrifier sa capacité prédictive.

Le déploiement du modèle via une application web interactive alimentée par Flask représente une étape cruciale pour rendre les résultats accessibles et utilisables. La sauvegarde du modèle avec Pickle garantit sa préservation et sa portabilité, facilitant ainsi son intégration dans divers contextes opérationnels sans nécessiter une reprise du processus d'entraînement. Cette démarche s'aligne avec l'objectif global de fournir une solution pratique et efficace pour la prédiction de l'espérance de vie, avec des implications potentielles pour les décideurs en santé publique et les professionnels de la santé au Maroc. En somme, ce projet témoigne de la puissance de l'analyse de données et du machine learning dans la compréhension et la prédiction des aspects complexes de la santé publique.