

PROYECTO INTEGRADOR - PREDICCIÓN DE LA ESPERANZA DE VIDA

ALEJANDRO ARANGO GIRALDO

Línea de énfasis en Ciencias de los Datos

ST1802 – Fundamentos de Ciencias de los Datos

CM0888 – Álgebra para Ciencias de los Datos

ST1800 – Almacenamiento y recuperación de la información

EC1801 – Estadística en analítica

**UNIVERSIDAD EAFIT
ESCUELA DE CIENCIAS APLICADAS E INGENIERÍA
CARRERA DE INGENIERÍA MECÁNICA
MEDELLÍN
2025 - 1**

Tabla de contenido

Introducción	6
Marco teórico.....	7
Metodología de investigación.....	8
1. Entendimiento del negocio	10
1.1 Determinación de los objetivos de negocio	10
1.1.1 Antecedentes.....	10
1.1.2 Objetivos de negocio.....	10
1.1.3 Criterios de Éxito	11
1.2 Evaluar la situación.....	11
1.2.1 Inventario de recursos	11
1.2.2 Requisitos, supuestos y restricciones.....	11
1.2.3 Riesgos y contingencias	12
1.3 Determinación de los objetivos de minería de datos	13
1.3.1 Objetivos de minería de datos	13
1.3.2 Criterios de éxito de la minería de datos	13
1.4 Producción del plan del proyecto	13
1.4.1 Plan del proyecto.....	13
1.5 Diccionario de datos	15
2. Entendimiento de datos	20
2.1 Importar librerías.....	20
2.2 Carga de datos	20
2.3 Análisis descriptivo	20
2.4 Calidad de datos	25
2.4.1 Completitud.....	25
2.4.2 Conformidad.....	28
2.4.3 Consistencia	31
2.4.4 Duplicidad.....	32
2.4.5 Integridad	32
2.4.6 Exactitud	32
3. Preparación de datos.....	33
3.1 Preprocesamiento de datos.....	33

3.1.1 Correcciones	33
3.1.2 Selección de filas y columnas.....	33
3.1.3 Imputación	35
3.1.4 Estandarización	41
3.1.5 Outliers	42
3.2 Preparación de características.....	46
3.3 Importancia de variables.....	47
3.3.1 Correlación de Pearson.....	47
3.3.2 Correlación de Spearman.....	49
3.3.3 Correlación parcial	50
3.3.4 Importancia con árboles de decisión.....	51
3.3.5 Valores SHAP	53
3.3.6 Selección de variables con Backward Selection.....	55
3.4 División de datos seleccionados en entrenamiento, validación y prueba.....	58
3.5 PCA.....	58
3.5.1 PCA para 2 componentes principales	58
3.5.2 PCA para 3 componentes principales	60
3.5.3 PCA para 6 componentes principales	61
3.5.4 División de componentes principales en entrenamiento, validación y prueba	61
4. Modelación	62
4.1 Multiple Linear Regression.....	62
4.1.1 Entrenamiento con todas las variables	63
4.1.2 Entrenamiento con variables seleccionadas.....	63
4.1.3 Entrenamiento con componentes principales	63
4.2 Lasso Regression.....	64
4.2.1 Entrenamiento con todas las variables	64
4.2.2 Entrenamiento con variables seleccionadas.....	64
4.2.3 Entrenamiento con componentes principales	65
4.3 Ridge Regression.....	65
4.3.1 Entrenamiento con todas las variables	66
4.3.2 Entrenamiento con variables seleccionadas.....	66
4.3.3 Entrenamiento con componentes principales	66
4.4 ElasticNet Regression	67

4.4.1 Entrenamiento con todas las variables	67
4.4.2 Entrenamiento con variables seleccionadas	68
4.4.3 Entrenamiento con componentes principales	68
4.5 Support Vector Regression	68
4.5.1 Entrenamiento con todas las variables	69
4.5.2 Entrenamiento con variables seleccionadas	69
4.5.3 Entrenamiento con componentes principales	70
4.6 K-Nearest Neighbors Regression	70
4.6.1 Entrenamiento con todas las variables	70
4.6.2 Entrenamiento con variables seleccionadas	71
4.6.3 Entrenamiento con componentes principales	71
4.7 Decision Tree	72
4.7.1 Entrenamiento con todas las variables	72
4.7.2 Entrenamiento con variables seleccionadas	73
4.7.3 Entrenamiento con componentes principales	73
4.8 Random Forest	74
4.8.1 Entrenamiento con todas las variables	74
4.8.2 Entrenamiento con variables seleccionadas	75
4.8.3 Entrenamiento con componentes principales	75
4.9 Gradient Boosting	76
4.9.1 Entrenamiento con todas las variables	77
4.9.2 Entrenamiento con variables seleccionadas	77
4.9.3 Entrenamiento con componentes principales	78
4.10 Selección del mejor modelo	79
4.11 Clasificación supervisada	79
4.11.1 Regresión Logística Multinomial	80
4.11.2 Selección del mejor modelo de clasificación supervisada	85
4.12 Clasificación no supervisada	85
5. Evaluación	93
5.1 Métricas MAE, MSE y MAPE	93
5.2 Análisis de desempeño	94
5.3 Intervalos de confianza para MAE, MSE y MAPE	102
5.3.1 Intervalo de confianza para MAE	103

5.3.2 Intervalo de confianza para MSE	104
5.3.3 Intervalo de confianza para MAPE	104
5.4 A/B testing.....	105
5.4.1 Modelo final vs Modelo de prueba.....	105
5.4.2 Modelo final vs Segundo modelo de Gradient Boosting	106
5.5 Evaluación del cumplimiento de los objetivos de negocio	106
5.6 Exportación del modelo final	107
6. Despliegue	108
Tecnología.....	110
Ciclo de vida de los datos y procesamiento analítico	110
1. Ambiente tecnológico.....	110
2. Origen de los datos.....	110
3. Ingesta.....	110
4. Almacenamiento	111
5. Procesamiento	111
6. Aplicaciones.....	112
Gráfico.....	112
Despliegue del modelo en un caso hipotético de Big Data	112
Expectativas analíticas	113
Aplicación del ciclo de vida.....	114
Comparación entre implementaciones	129
Conclusiones	130
Reconocimientos	131
Referencias.....	131

Introducción

En el marco del desarrollo de soluciones basadas en ciencia de datos aplicadas a problemáticas del ámbito de la salud pública, el presente proyecto tiene como propósito principal el diseño e implementación de un modelo predictivo capaz de estimar la esperanza de vida de un país a partir de variables socioeconómicas, sanitarias y demográficas. Este trabajo se inscribe dentro de un ejercicio académico de la línea de énfasis en ciencia de datos de la Universidad EAFIT, y se fundamenta en fuentes de datos oficiales provistas por la Organización Mundial de la Salud (OMS) y la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO).

La motivación de este estudio surge de la necesidad de comprender en mayor profundidad cómo diversos factores —especialmente aquellos relacionados con la inmunización, el desarrollo humano, el gasto en salud y la prevalencia de enfermedades— inciden en la longevidad de las poblaciones. Este proyecto busca subsanar vacíos como el uso de datos de un único año o la omisión de variables relevantes como las tasas de vacunación, mediante el análisis de un conjunto de datos que abarca el periodo 2000–2016 y contempla 32 variables de 183 países.

Desde una perspectiva metodológica, se adopta el enfoque CRISP-DM, lo que permite estructurar el proceso analítico en etapas claramente definidas que incluyen el entendimiento del negocio, exploración y preparación de los datos, modelado, evaluación, despliegue, y comunicación de resultados. Se plantea como objetivo de negocio apoyar la toma de decisiones estratégicas orientadas a mejorar la salud y longevidad de la población, mediante la identificación de los factores más influyentes en la esperanza de vida y la formulación de recomendaciones basadas en evidencia.

En cuanto a los objetivos técnicos, se pretende desarrollar un modelo de regresión con un error absoluto medio (MAE) mínimo, utilizando técnicas avanzadas de análisis exploratorio, selección de características, y modelado predictivo. Igualmente, se explorarán modelos de clasificación supervisada y no supervisada para enriquecer la

investigación. Se espera que los resultados de este análisis no solo tengan valor académico, sino que también aporten información útil y accionable para el diseño de políticas públicas más efectivas por parte de gobierno y organizaciones no gubernamentales.

Marco teórico

La esperanza de vida ha sido uno de los indicadores más utilizados en el estudio del bienestar social desde las primeras aproximaciones estadísticas al fenómeno poblacional. En sus orígenes, la construcción de tablas de vida por parte de John Graunt en el siglo XVII y, posteriormente, las generalizaciones matemáticas realizadas por Edmond Halley, representaron las primeras tentativas sistemáticas de medir la duración promedio de la vida humana en contextos urbanos europeos (Graunt, 1662; Halley, 1693). Estas aproximaciones dieron paso a una larga tradición demográfica en la que la esperanza de vida ha sido entendida no solo como un reflejo de las condiciones sanitarias, sino también como un resultado de variables estructurales tales como el ingreso, la educación, la alimentación y la estabilidad política.

Durante el siglo XX, autores como Thomas McKeown (1976) pusieron de relieve que el incremento de la esperanza de vida observado desde el siglo XIX no obedecía exclusivamente a intervenciones médicas, sino que estaba profundamente relacionado con las mejoras en las condiciones socioeconómicas, especialmente la nutrición y el saneamiento básico. Este enfoque marcó un punto de inflexión en la forma de comprender los determinantes de la salud poblacional, sentando las bases de lo que más tarde se conocería como los “determinantes sociales de la salud”. A partir de estas ideas, Michael Marmot desarrolló un marco conceptual integral en el que la salud y la longevidad son vistas como productos de desigualdades estructurales, con impacto acumulativo desde el nacimiento hasta la muerte (Marmot, 2005; 2017).

Hoy día, este indicador, también conocido como esperanza de vida al nacer, se ha acotado a la siguiente definición: el número promedio de años que una persona puede esperar vivir desde el nacimiento (OMS, 2024). Este indicador constituye una métrica esencial del desarrollo humano y de la efectividad de las políticas públicas en salud, educación y bienestar social. De acuerdo con el Programa de las Naciones Unidas para el Desarrollo (PNUD), la esperanza de vida es un componente esencial en el cálculo del Índice de Desarrollo Humano (IDH), el cual es una medida resumen del logro promedio en dimensiones clave del desarrollo humano: una vida larga y saludable, estar informado y tener un nivel de vida decente. El IDH es la media geométrica de los índices normalizados para cada una de las tres dimensiones. (PNUD, 2024).

Factores como la calidad y cobertura del sistema de salud, condiciones socioeconómicas y medioambientales, nivel educativo, prevalencia de enfermedades transmisibles y no transmisibles, entre otros, inciden directa o indirectamente en la esperanza de vida. El avance de la ciencia de datos ha permitido un análisis más profundo y dinámico de estas variables y su relación con la esperanza de vida, empleando herramientas computacionales que identifican patrones complejos en grandes volúmenes de información. En este contexto, el uso de aprendizaje automático (machine learning) se ha consolidado como un enfoque eficaz para modelar y predecir variables dependientes de múltiples factores.

Adicionalmente, gracias a la mejora en las tecnologías digitales y el acceso a grandes volúmenes de datos sanitarios, educativos y económicos, surgió una nueva generación de investigaciones orientadas a modelar y predecir la esperanza de vida desde un enfoque computacional. El uso de bases de datos internacionales estandarizadas, como las ofrecidas por el Global Health Observatory de la OMS y el Institute for Statistics de la UNESCO, permitió combinar múltiples dimensiones del desarrollo humano y sanitario. A partir de estos datos, se han desplegado enfoques metodológicos más sofisticados basados en aprendizaje automático, lo cual ha permitido superar muchas de las limitaciones de los modelos lineales clásicos, como la rigidez de los supuestos estadísticos, la linealidad de las relaciones y la dificultad para capturar efectos combinados o no evidentes entre variables.

Estudios como el de Kontis et al. (2017), que emplearon modelos bayesianos para predecir la esperanza de vida en países de la OCDE, demostraron que enfoques probabilísticos pueden proyectar con alta precisión la evolución demográfica futura bajo distintos escenarios. Sin embargo, su aplicabilidad fue limitada en contextos con datos incompletos o con alta variabilidad contextual. En contraste, los modelos de *machine learning*, particularmente los algoritmos de árboles de decisión como *Random Forest* y *XGBoost*, han mostrado un mejor desempeño en términos de error de predicción, sensibilidad a relaciones no lineales y robustez ante datos faltantes (Lipesa et al., 2023; Pooja & Archana, 2023).

Alanazi (2022) demostró que la integración de técnicas de aprendizaje supervisado con indicadores sociales, sanitarios y económicos permite alcanzar una precisión sin precedentes en la predicción de la esperanza de vida, especialmente cuando se emplean algoritmos de tipo *ensemble*. En la misma línea, estudios de salud pública en países de bajos ingresos han revelado que la esperanza de vida está fuertemente influenciada por variables como la cobertura de vacunación infantil, el gasto público en salud como porcentaje del PIB, la prevalencia de VIH y la alfabetización (Wang et al., 2016; WHO, 2023). Estas correlaciones han sido replicadas mediante técnicas de selección de características que priorizan variables según su impacto marginal en el modelo, utilizando herramientas como SHAP (Lundberg & Lee, 2017), que permiten interpretar los modelos complejos y aumentar su utilidad práctica para tomadores de decisión.

Uno de los aportes más relevantes de la literatura reciente ha sido el paso de la predicción a la explicación mediante el uso de inteligencia artificial explicable (*XAI*). La posibilidad de descomponer una predicción en los factores específicos que más contribuyen a ella representa un avance sustantivo frente a la opacidad de los modelos tipo caja negra, lo cual ha sido identificado como una barrera ética y operativa en contextos de política pública (Markus et al., 2021). Esta línea de trabajo ha sido fundamental para que los sistemas de predicción basados en inteligencia artificial puedan ser utilizados como soporte a decisiones gubernamentales, respetando los principios de transparencia, trazabilidad y rendición de cuentas.

Sin embargo, a pesar del notable progreso metodológico, existen importantes vacíos en la literatura. En primer lugar, pocos estudios han logrado integrar de manera simultánea variables sanitarias, económicas y educativas en modelos de alta precisión y explicabilidad. En segundo lugar, la mayoría de los modelos han sido desarrollados para contextos nacionales específicos, lo cual limita su capacidad de generalización. En tercer lugar, aún es incipiente el desarrollo de herramientas prácticas derivadas de estos modelos, como *dashboards* interactivos o aplicaciones web, que faciliten el uso de estas predicciones por parte de instituciones gubernamentales, organismos multilaterales y organizaciones no gubernamentales. En este sentido, el presente trabajo se inscribe dentro de una línea de investigación emergente que busca, a partir de datos globales estandarizados y técnicas de aprendizaje automático interpretables, construir un modelo predictivo robusto, transparente y operativamente útil para apoyar la toma de decisiones en salud pública a escala global.

Metodología de investigación

La metodología empleada para este proyecto se basa en el enfoque CRISP-DM (Cross-Industry Standard Process for Data Mining), omitiendo la etapa inicial de entendimiento del negocio y la etapa final de despliegue. La primera puede considerarse satisfecha en la formulación de este anteproyecto, y la última será un incremental en caso de contar con el tiempo y los recursos suficientes. Con esto en mente, para dar cumplimiento al objetivo general se contemplan las siguientes etapas dentro de la metodología de investigación: entendimiento de datos, preparación de datos, modelación y evaluación.

En la etapa de **entendimiento de datos**, se importarán las librerías, se cargarán los datos, se desarrollará un análisis descriptivo derivado de los resultados de las técnicas: `head()`, `tail()`, `info()`, y `describe()`, y, finalmente, se evaluará la calidad de los datos según los criterios de completitud, conformidad, consistencia, duplicidad, integridad, y exactitud.

En la etapa de **preparación de datos**, se harán las correcciones y selección de columnas y filas a partir de los resultados de la etapa anterior. Se implementará una combinación de interpolación, NOCB (Next Observation Carried Backward), y LOCF (Last Observation Carried Forward) para imputar los datos faltantes. Se estandarizarán los datos y, posteriormente, se detectarán outliers mediante tres métodos: distancia de Mahalanobis, DBSCAN clustering, y boxplots. Se prepararán las características aplicando one-hot encoding a las variables categóricas seleccionadas, y se eliminarán las variables que ya no sean necesarias tras los análisis previos.

Se valorará la importancia de las variables con análisis de correlación (Pearson, Spearman, Correlación parcial), árboles de decisión (importancia de características en un modelo de Gradient Boosting), y valores SHAP. Con base a este análisis, se empleará el método de *Backward Selection* con un modelo de Gradient Boosting, en el cual, en cada iteración, se eliminará una variable de poca importancia y se determinará el desempeño del modelo frente a este cambio. Se iterará hasta encontrar un cambio significativo en el desempeño del modelo, indicando entonces qué variables pueden ser eliminadas sin afectar negativamente los resultados. Se hará un análisis de componentes principales (PCA), iterando hasta encontrar la cantidad de componentes principales que expliquen mínimamente el 70% de la varianza de los datos. Finalmente, se construirán los conjuntos de entrenamiento, validación y prueba, tanto para los resultados del *Backward Selection* como para los del *PCA* con la siguiente proporción de datos respectivamente: 60/20/20.

En la etapa de **modelación**, se entrenarán y compararán los siguientes 9 modelos de regresión: lineal múltiple, Lasso, Ridge, ElasticNet, SVR (regresión de vectores de soporte), KNN (regresión de k-vecinos más cercanos), árbol de decisión, random forest, y gradient boosting, cada una en tres configuraciones distintas: con todas las variables, con las variables seleccionadas del *Backward Selection*, y con los componentes principales obtenidos mediante *PCA*, para un total de 27 combinaciones. Para cada una de las combinaciones, se utilizará la técnica de RandomizedSearchCV para obtener los mejores parámetros, se entrenará el modelo sobre estos parámetros, y se evaluarán las métricas de MAE, MSE, MAPE, y Score. Con base en las métricas, se seleccionará el mejor modelo.

De igual forma, se explorarán los modelos de regresión logística multinomial para clasificación supervisada y KMeans para clasificación no supervisada. Para el primero se entrenarán 3 modelos con los 3 conjuntos de entrenamiento utilizados para los modelos de regresión y empleando la técnica de RandomizedSearchCV para la optimización de hiperparámetros. La evaluación se hará sobre las métricas de precision, recall, f1-score y accuracy. El segundo buscará segmentar los países según sus características del 2016 para después comparar los clusters generados con la esperanza de vida real para este mismo año.

En la etapa de **evaluación**, se tomará el modelo obtenido previamente y se reentrenará con la combinación de los conjuntos de entrenamiento y validación, y se reevaluará su desempeño según las métricas de MAE, MAPE y MSE. Se realizará un análisis de desempeño del modelo sobre el conjunto de prueba para verificar el comportamiento de las predicciones y los errores en diferentes escenarios (distribución del error absoluto, Predicción vs. Real, error absoluto y MAE vs. Variables). Para ello, se emplearán valores SHAP, técnicas de visualización como gráficos de barras, dispersión, de caja, etc. Se calculará el intervalo de confianza del 95% para las métricas MAE, MAPE y MSE empleando la técnica de Bootstrap Percentile Confidence Intervals. Se ejecutará una prueba A/B con el Wilcoxon signed rank test para comparar el modelo final con un modelo cuyas predicciones siempre serán la media simple de la variable de salida. De esta manera, se puede verificar que el modelo elegido si logre superar significativamente la media simple, y con qué nivel de confianza se puede asegurar esto. Finalmente, se evaluará el cumplimiento de los objetivos, se reentrenará el modelo en la totalidad de los datos, y se exportará para su futuro despliegue.

Por último, en la etapa de **despliegue** se propondrá una aplicación simple con la librería tkinter que permita al usuario ingresar sus datos y generar predicciones sobre ellos, brindando a su vez un gráfico de los valores SHAP para indicar la importancia de las variables en la predicción y aportar información clave para la toma de

decisiones. Asimismo, se planteará un ciclo de vida de los datos para un despliegue en un ambiente productivo real, estableciendo dos alternativas de desarrollo que varían en complejidad. Una hace uso, en su totalidad, de servicios ofrecidos por AWS para un desarrollo en la nube, y el otro hace uso parcial de estos servicios, empleando únicamente el servicio de AWS S3 para el almacenamiento de la información. A manera de demo, se implementa el proyecto en la interfaz web de AWS Academy Learner Lab, adaptándolo a un caso de Big Data, y comparando los resultados con la implementación original.

1. Entendimiento del negocio

1.1 Determinación de los objetivos de negocio

1.1.1 Antecedentes

La Organización Mundial de la Salud (OMS), fundada en 1948, es la autoridad directiva y coordinadora en asuntos de sanidad internacional en el sistema de las Naciones Unidas. Estos prestan apoyo a los países en la coordinación de las actividades de diferentes sectores del gobierno y de los asociados –incluidos asociados bilaterales y multilaterales, fondos y fundaciones, organizaciones de la sociedad civil y el sector privado– para que logren sus objetivos sanitarios y para apoyar sus estrategias y políticas sanitarias nacionales (OMS, 2025).

La OMS coordina la labor sanitaria internacional promoviendo la colaboración por medio de la movilización de alianzas y de diferentes agentes del ámbito de la salud. Sus ámbitos de actividad principales son: enfermedades no transmisibles, enfermedades transmisibles, preparación, vigilancia y respuesta a las crisis, promoción de la salud a lo largo del ciclo de vida, sistemas de salud, y servicios institucionales (OMS, 2025).

En el pasado se han realizado abundantes estudios sobre los factores que afectan la esperanza de vida, considerando variables demográficas, composición de ingresos y tasas de mortalidad. Sin embargo, la problemática radica en que estos estudios no consideraron el efecto de la inmunización y el índice de desarrollo humano. Igualmente, algunas de las investigaciones anteriores se realizaron considerando únicamente un conjunto de datos de un año para todos los países.

Para resolver esta problemática se propone un análisis que considere datos de un período de 2000 a 2016 para 183 países. Asimismo, se considerarán vacunas importantes como la hepatitis B, la polio y la difteria. Se combinan entonces repositorios de datos del Observatorio Mundial de la Salud (WHO por sus siglas en inglés) y de la UNESCO (United Nations Educational Scientific and Culture Organization), para obtener un conjunto de datos que englobe los factores representativos que aportaron a la mejora de las tasas de mortalidad humana en los últimos años.

Con este dataset, se busca realizar un análisis exhaustivo que permita identificar como factores de inmunización, económicos, sociales y otros, influyen en la esperanza de vida de la población de un país. Se espera que el estudio ayude a identificar qué áreas son más críticas para los países que buscan aumentar su esperanza de vida. Así como servir de base fundacional para la toma de decisiones informadas y proposición de proyectos que lleven a esto.

1.1.2 Objetivos de negocio

Objetivo Principal

Desarrollar un modelo predictivo que permita estimar la esperanza de vida de un determinado país a partir de características propias de este y de su población.

Preguntas de Negocio

- ¿Cómo se ve afectada la esperanza de vida por los hábitos alimentarios, estilo de vida, ejercicio, y consumo de alcohol de la población?

- ¿Los países densamente poblados tienden a tener una esperanza de vida más baja?
- ¿Cuál es el impacto de la cobertura de vacunación (como Hepatitis B, Polio, Difteria, entre otras) en la esperanza de vida?
- ¿Qué efecto tienen las variables económicas como la Renta Nacional Bruta (RNB) per capita, medida en dólares estadounidenses y el porcentaje de gasto en salud sobre la esperanza de vida?
- ¿Qué influencia tiene el continente o la región geográfica en la esperanza de vida, y cómo se manifiestan las diferencias entre continentes?
- ¿Hasta qué punto afectan las enfermedades infecciosas como el VIH/SIDA a la esperanza de vida en diferentes países?

1.1.3 Criterios de Éxito

Identificar los factores más significativos que afectan la esperanza de vida para proporcionar recomendaciones concretas a la OMS y a los gobiernos sobre dónde enfocar los esfuerzos para aumentar la longevidad y el bienestar de la población.

Indicar posibles cambios en áreas de mejora que, al ser aplicadas, resulten en el aumento de la predicción de la esperanza de vida de un determinado país.

1.2 Evaluar la situación

1.2.1 Inventario de recursos

Personal

Estudiante de 10.^º semestre de ingeniería mecánica de la universidad EAFIT con énfasis en ciencia de datos.

Datos

1. **Dataset de la OMS:** conjunto de datos relacionados con la esperanza de vida, factores de salud, y algunos aspectos socioeconómicos de 183 países entre los años 2000 y 2016 obtenido del repositorio de datos del Global Health Observatory (GHO) de la Organización Mundial de la Salud (OMS).
2. **Dataset de la UNESCO:** conjunto de datos relacionado con aspectos socioeconómicos y algunos factores de salud de 183 países entre los años 2000 y 2016.

El archivo combinado final (conjunto de datos final) consta de 32 columnas y 3111 filas.

Recursos informáticos

- **Plataformas de hardware:** Infraestructura tecnológica con capacidad suficiente para procesar y analizar grandes conjuntos de datos y entrenar y desplegar modelos de ML complejos.
- **Software de análisis de datos:** Lenguajes especializados para análisis estadístico y modelado predictivo, como Python, y ambiente de programación interactivo como Jupyter Notebooks.
- **Herramientas de visualización de datos:** Software para crear gráficos y visualizaciones para el análisis y la presentación de resultados, como Power BI, o bibliotecas de uso público como Matplotlib o Seaborn.
- **Herramienta de apoyo:** LLMs como Chat GPT versión 4o para el apoyo en la generación de código y consulta de información que será debidamente revisada y modificada a discreción.

1.2.2 Requisitos, supuestos y restricciones

Requisitos

- **Cronograma de finalización:** El proyecto debe completarse para el 3 de junio de 2025 y debe presentarse ante el representante de la OMS este mismo día.
- **Calidad y comprensibilidad de los resultados:** Se deben llevar a cabo todos los pasos de preprocesamiento de los datos, incluyendo la verificación de las dimensiones de calidad del dataset para cerciorarse que los resultados sean correctos y relevantes. De igual forma, se debe documentar

adecuadamente todas las etapas del proyecto, de manera que los resultados y el proceso para llegar a estos sea comprensible para los clientes.

- **Permisos de uso de los datos:** El dataset empleado en el proyecto es la agrupación de otros datasets de libre acceso provenientes de la OMS y la UNESCO. Estos fueron recopilados y combinados por el usuario de Kaggle mmattson, y puestos a disposición de la comunidad de Kaggle el año 2020 en el siguiente link: <https://www.kaggle.com/datasets/mmattson/who-national-life-expectancy/data>

Supuestos

- **Precisión del dataset:** Se asume que los datos registrados por la OMS y la UNESCO son precisos y están libres de errores significativos, y que el dataset publicado en Kaggle conserva los datos originales de estas fuentes sin ninguna modificación. Cualquier corrección de datos se documentará y justificará.
- **Representatividad del dataset:** Se asume que el dataset es representativo de los 183 países incluidos en este y que la información es verídica y fue recolectada por medios certeros.

Restricciones

- **Tamaño del dataset:** El tamaño del conjunto de datos es una restricción para el entrenamiento de modelos de ML, especialmente para modelos complejos. Por un lado, el dataset solo tiene 3111 registros, de los cuales, hay datos faltantes. Por otra parte, aunque tiene una cantidad adecuada de variables para la predicción, hay información adicional que podría resultar beneficiosa para el entrenamiento del modelo predictivo, pero que no fue incluida en el dataset.
- **Acceso a datos en tiempo real:** El dataset utilizado abarca el período de 2000 a 2016, por lo que no se tiene acceso a datos en tiempo real ni a datos más recientes. Esto limita considerablemente el alcance de los resultados, derivando en información que puede resultar antiguada para el contexto actual.

1.2.3 Riesgos y contingencias

1. **Riesgos relacionados con la precisión de datos:**
 - **Datos incorrectos o faltantes:** Existe el riesgo de que el dataset contenga errores, datos faltantes o inconsistencias. Esto podría afectar la precisión del análisis y los resultados.
 - **Contingencia:** Implementar procesos de limpieza y verificación de datos antes del análisis. Se pueden utilizar técnicas de imputación para abordar datos faltantes y validación cruzada para verificar la precisión de los datos.
2. **Riesgos relacionados con el cronograma:**
 - **Retrasos en el proyecto:** El cronograma podría verse afectado por factores imprevistos.
 - **Contingencia:** Establecer plazos intermedios y puntos de control para asegurar el progreso constante.
3. **Riesgos relacionados con la comprensión de resultados:**
 - **Dificultades para interpretar el análisis:** Los resultados del proyecto podrían ser complejos o difíciles de entender para las partes interesadas no técnicas.
 - **Contingencia:** Desarrollar un informe claro, con visualizaciones y explicaciones detalladas de los resultados.
4. **Riesgos relacionados con la representatividad de resultados:**
 - **Resultados poco representativos o dramáticos:** Existe el riesgo de que el análisis produzca resultados que no sean suficientemente representativos del conjunto completo de datos o de la realidad de los países analizados. Esto podría llevar a interpretaciones erróneas o a conclusiones extremas, afectando la utilidad del proyecto y las decisiones basadas en los resultados.
 - **Contingencia:** Implementar técnicas de validación y evaluación cruzada para asegurar la representatividad de los resultados. Utilizar múltiples modelos y enfoques para verificar la coherencia de los resultados. Incluir medidas estadísticas como el intervalo de confianza para dar contexto a las predicciones y evitar conclusiones precipitadas.

1.3 Determinación de los objetivos de minería de datos

1.3.1 Objetivos de minería de datos

1. Determinar patrones, correlaciones y tendencias entre las diferentes variables y la esperanza de vida.
2. Desarrollar un modelo predictivo que permita estimar la esperanza de vida de un país con base en factores clave relacionados con inmunizaciones, mortalidad, economía, sociedad y otros indicadores de salud, cuyo MAE sea el mínimo posible.

1.3.2 Criterios de éxito de la minería de datos

1. Desarrollo de un modelo predictivo de regresión con el mínimo MAE sobre el conjunto de prueba entre distintos modelos entrenados.
2. Selección del mejor modelo a partir de los errores absolutos con mínimo un 95% de confianza.

1.4 Producción del plan del proyecto

1.4.1 Plan del proyecto

Las etapas del proyecto serán las propuestas por la metodología CRISP-DM.

1. Entendimiento del negocio
 - 1.1. Determinación de los objetivos de negocio
 - 1.1.1. Antecedentes
 - 1.1.2. Objetivos y preguntas de negocio
 - 1.1.3. Criterios de éxito
 - 1.2. Evaluar la situación
 - 1.2.1. Inventario de recursos
 - 1.2.2. Requisitos, supuestos y restricciones
 - 1.2.3. Riesgos y contingencias
 - 1.3. Determinación de los objetivos de minería de datos
 - 1.3.1. Objetivos de minería de datos
 - 1.3.2. Criterios de éxito de la minería de datos
 - 1.4. Producción del plan del proyecto
2. Entendimiento de datos
 - 2.1. Importar librerías
 - 2.2. Carga de datos
 - 2.3. Análisis descriptivo
 - 2.4. Calidad de datos
 - 2.4.1. Completitud
 - 2.4.2. Conformidad
 - 2.4.3. Consistencia
 - 2.4.4. Duplicidad
 - 2.4.5. Integridad
 - 2.4.6. Exactitud
3. Preparación de datos
 - 3.1. Preprocesamiento de datos
 - 3.1.1. Correcciones
 - 3.1.2. Selección de filas y columnas

- 3.1.3. Imputación
 - 3.1.4. Estandarización
 - 3.1.5. Outliers
 - 3.1.5.1. Distancia de Mahalanobis
 - 3.1.5.2. DBSCAN Clustering
 - 3.1.5.3. Boxplots
 - 3.2. Preparación de características
 - 3.3. Importancia de variables
 - 3.3.1. Correlación de Pearson
 - 3.3.2. Correlación de Spearman
 - 3.3.3. Correlación parcial
 - 3.3.4. Importancia con árboles de decisión
 - 3.3.4.1. División de datos en entrenamiento, validación y prueba
 - 3.3.4.2. Entrenamiento del modelo
 - 3.3.4.3. Importancia de las características
 - 3.3.5. Valores SHAP
 - 3.3.6. Selección de variables con Backward Selection
 - 3.3.6.1. Iteración 1
 - 3.3.6.2. Iteración 2
 - 3.3.6.3. Iteración 3
 - 3.3.6.4. Iteración 4
 - 3.3.6.5. Iteración 5
 - 3.3.6.6. Iteración 6
 - 3.3.6.7. Iteración 7
 - 3.4. División de datos seleccionados en entrenamiento, validación y prueba
 - 3.5. PCA
 - 3.5.1. PCA para 2 componentes principales
 - 3.5.2. PCA para 3 componentes principales
 - 3.5.3. PCA para 6 componentes principales
 - 3.5.4. División de componentes principales en entrenamiento, validación y prueba
- 4. Modelación
 - 4.1. Multiple Linear Regression
 - 4.1.1. Entrenamiento con todas las variables
 - 4.1.2. Entrenamiento con variables seleccionadas
 - 4.1.3. Entrenamiento con componentes principales
 - 4.2. Lasso Regression
 - 4.2.1. Entrenamiento con todas las variables
 - 4.2.2. Entrenamiento con variables seleccionadas
 - 4.2.3. Entrenamiento con componentes principales
 - 4.3. Ridge Regression
 - 4.3.1. Entrenamiento con todas las variables
 - 4.3.2. Entrenamiento con variables seleccionadas
 - 4.3.3. Entrenamiento con componentes principales
 - 4.4. ElasticNet Regression
 - 4.4.1. Entrenamiento con todas las variables
 - 4.4.2. Entrenamiento con variables seleccionadas
 - 4.4.3. Entrenamiento con componentes principales

- 4.5. Support Vector Regression
 - 4.5.1. Entrenamiento con todas las variables
 - 4.5.2. Entrenamiento con variables seleccionadas
 - 4.5.3. Entrenamiento con componentes principales
- 4.6. K-Nearest Neighbors Regression
 - 4.6.1. Entrenamiento con todas las variables
 - 4.6.2. Entrenamiento con variables seleccionadas
 - 4.6.3. Entrenamiento con componentes principales
- 4.7. Decision Tree
 - 4.7.1. Entrenamiento con todas las variables
 - 4.7.2. Entrenamiento con variables seleccionadas
 - 4.7.3. Entrenamiento con componentes principales
- 4.8. Random Forest
 - 4.8.1. Entrenamiento con todas las variables
 - 4.8.2. Entrenamiento con variables seleccionadas
 - 4.8.3. Entrenamiento con componentes principales
- 4.9. Gradient Boosting
 - 4.9.1. Entrenamiento con todas las variables
 - 4.9.2. Entrenamiento con variables seleccionadas
 - 4.9.3. Entrenamiento con componentes principales
- 4.10. Selección del mejor modelo de regresión
- 4.11. Clasificación supervisada
 - 4.11.1. Regresión Logística Multinomial
 - 4.11.1.1. Entrenamiento con todas las variables
 - 4.11.1.2. Entrenamiento con variables seleccionadas
 - 4.11.1.3. Entrenamiento con componentes principales
 - 4.11.2. Selección de mejor modelo de clasificación supervisada
- 4.12. Clasificación no supervisada
- 5. Evaluación
 - 5.1. Métricas MAE, MSE y MAPE
 - 5.2. Análisis de desempeño
 - 5.3. Intervalos de confianza para MAE, MSE y MAPE
 - 5.3.1. Intervalos de confianza para MAE
 - 5.3.2. Intervalos de confianza para MSE
 - 5.3.3. Intervalos de confianza para MAPE
 - 5.4. A/B testing
 - 5.4.1. Modelo final vs Modelo de prueba
 - 5.4.2. Modelo final vs Segundo modelo de Gradient Boosting
 - 5.5. Evaluación del cumplimiento de los objetivos de negocio
 - 5.6. Exportación del modelo final
- 6. Despliegue

1.5 Diccionario de datos

Nombre	Tipo	Tamaño	Descripción	Posibles valores
--------	------	--------	-------------	------------------

country	Cadena	45	Nombre del país al que pertenece la observación	Angola, Burundi, Benin, Burkina Faso, Botswana, Central African Republic, United Republic of Tanzania, Uganda, South Africa, Zambia, Zimbabwe, South Sudan Sao Tome and Principe, Eswatini, Seychelles, Chad, Togo, Namibia, Niger, Nigeria, Rwanda, Senegal, Sierra Leone, Madagascar, Mali, Mozambique, Mauritania, Mauritius, Malawi, Gambia, Guinea-Bissau, Equatorial Guinea, Kenya, Liberia, Lesotho, Algeria, Eritrea, Ethiopia, Gabon, Ghana, Guinea, Côte d'Ivoire, Cameroon, Democratic Republic of the Congo, Congo, Comoros, Cabo Verde, Argentina, Antigua and Barbuda, Bahamas, Belize, United States of America, Saint Vincent and the Grenadines, Venezuela (Bolivarian Republic of), Paraguay, El Salvador, Suriname, Trinidad and Tobago, Uruguay, Saint Lucia, Mexico, Nicaragua, Panama, Peru, Grenada, Guatemala, Guyana, Honduras, Haiti, Jamaica, Costa Rica, Cuba, Dominican Republic, Ecuador, Bolivia (Plurinational State of), Brazil, Barbados, Canada, Chile, Colombia, Afghanistan, United Arab Emirates, Bahrain, Djibouti, Egypt, Iran (Islamic Republic of), Somalia, Syrian Arab Republic, Tunisia, Yemen, Oman, Pakistan, Qatar, Saudi Arabia, Sudan, Iraq, Jordan, Kuwait, Lebanon, Libya, Morocco, Albania, Armenia, Austria, Azerbaijan, Belgium, Sweden, Tajikistan, Turkmenistan, Turkey, Ukraine, Uzbekistan, Serbia, Slovakia, Slovenia, Netherlands, Norway, Poland, Portugal, Romania, Russian Federation, Latvia, Republic of Moldova, Republic of North Macedonia, Malta, Montenegro, Italy, Kazakhstan, Kyrgyzstan, Lithuania, Luxembourg, Greece, Croatia, Hungary, Ireland, Iceland, Israel, Spain, Estonia, Finland, France, United Kingdom of Great Britain and Northern Ireland, Georgia, Czechia, Germany, Denmark, Bulgaria, Bosnia and Herzegovina, Belarus, Switzerland, Cyprus, Bangladesh, Bhutan, Indonesia, India, Sri Lanka, Maldives, Myanmar, Nepal, Democratic Peoples Republic of Korea, Thailand, Timor-Leste, Australia, Brunei Darussalam, China, Fiji, Micronesia (Federated States of), Tonga, Viet Nam, Vanuatu, Samoa, Philippines, Papua New Guinea, Singapore, Solomon Islands, Mongolia, Malaysia, New Zealand, Japan, Cambodia, Kiribati, Republic of Korea, Lao Peoples Democratic Republic
country_code	Cadena	3	Identificador del país	AGO, BDI, BEN, BFA, BWA, CAF, TZA, UGA, ZAF, ZMB, ZWE, SSD, STP, SWZ, SYC, TCD, TGO, NAM, NER, NGA, RWA, SEN, SLE, MDG, MLI, MOZ, MRT, MUS, MWI, GMB, GNB, GNQ, KEN, LBR, LSO, DZA, ERI, ETH, GAB, GHA, GIN, CIV, CMR, COD, COG, COM, CPV, ARG, ATG, BHS, BLZ, USA, VCT, VEN, PRY, SLV, SUR, TTO, URY, LCA, MEX, NIC, PAN, PER, GRD, GTM, GUY, HND, HTI, JAM, CRI, CUB, DOM, ECU, BOL, BRA, BRB, CAN, CHL, COL, AFG, ARE, BHR, DJI, EGY, IRN, SOM, SYR, TUN, YEM, OMN, PAK, QAT, SAU, SDN, IRQ, JOR, KWT, LBN, LBY, MAR, ALB, ARM, AUT, AZE, BEL, SWE, TJK, TKM, TUR, UKR, UZB, SRB, SVK, SVN, NLD, NOR, POL, PRT, ROU, RUS, LVA, MDA, MKD, MLT, MNE, ITA, KAZ, KGZ, LTU, LUX, GRC, HRV, HUN, IRL, ISL, ISR, ESP, EST, FIN, FRA, GBR, GEO, CZE, DEU, DNK, BGR, BIH, BLR, CHE, CYP, BGD, BTN, IDN, IND, LKA, MDV, MMR, NPL, PRK, THA, TLS, AUS, BRN, CHN, FJI, FSM, TON, VNM, VUT, WSM, PHL, PNG, SGP, SLB, MNG, MYS, NZL, JPN, KHM, KIR, KOR, LAO
region	Cadena	20	Región global a la que pertenece el país	Europe, Africa, Americas, Eastern Mediterranean, Western Pacific, South-East Asia

year	Entero	4	Año en el que se recopilaron los datos para la observación	Entero entre 2000 y 2016
life_expect	Flotante	17	Esperanza de vida media al nacer, medida en años	Flotante mayor a 0
life_exp60	Flotante	17	Esperanza de vida media a los 60 años, medida en años	Flotante mayor a 0
adult_mortality	Flotante	7	Número de muertes de adultos entre los 15 y 60 años, por cada 1000 habitantes	Flotante entre 0 y 1000
infant_mort	Flotante	18	Tasa de mortalidad de niños menores de un año por cada 1000 nacimientos vivos	Flotante entre 0 y 1000
age1-4mort	Flotante	19	Tasa de mortalidad de niños entre 1 y 4 años de edad por cada 1000 nacimientos vivos	Entero entre 0 y 1000
alcohol	Flotante	17	Consumo anual de alcohol por persona mayor de 15 años, medido en litros de alcohol puro	Flotante mayor a 0
bmi	Flotante	3	Índice de Masa Corporal promedio de toda la población	Flotante mayor a 0
age5-19thinness	Flotante	3	Porcentaje de niños de 5 a 19 años que se consideran delgados según el Índice de Masa Corporal (BMI < mediana - 2 d.s)	Flotante entre 0 y 100
age5-19obesity	Flotante	3	Porcentaje de niños y adolescentes de 5 a 19 años que se consideran obesos según el Índice de Masa Corporal (BMI > mediana + 2 d.s)	Flotante entre 0 y 100
hepatitis	Flotante	3	Porcentaje de niños de un año que han recibido la vacuna contra la hepatitis B	Flotante entre 0 y 100

measles	Flotante	3	Porcentaje de niños de un año que han recibido la vacuna contra el sarampión (MCV1)	Flotante entre 0 y 100
polio	Flotante	3	Porcentaje de niños de un año que han recibido la vacuna contra la polio	Flotante entre 0 y 100
diphtheria	Flotante	3	Porcentaje de niños de un año que han recibido la vacuna DTP3 (difteria, tétanos y tosferina)	Flotante entre 0 y 100
basic_water	Flotante	17	Porcentaje de la población usando por lo menos servicios básicos de agua potable	Flotante entre 0 y 100
doctors	Flotante	17	Número de doctores por cada 10000 habitantes	Flotante entre 0 y 10000
hospitals	Flotante	18	Densidad total de hospitales por cada 100000 habitantes	Flotante entre 0 y 100000
gni_capita	Flotante	7	Renta Nacional Bruta (RNB) per capita, medida en dólares estadounidenses	Flotante mayor a 0
gghe-d	Flotante	17	Gasto interno general del gobierno en salud (GGHE-D) como porcentaje del producto interno bruto (PIB)	Flotante entre 0 y 100
che_gdp	Flotante	17	Gasto actual en salud (CHE) como porcentaje del producto interno bruto (PIB)	Flotante entre 0 y 100
une_pop	Flotante	10	Población del país en miles	Flotante mayor a 0
une_infant	Flotante	3	Número de muertes de niños menores de 1 año por cada 1000 nacimientos vivos	Flotante entre 0 y 1000

une_life	Flotante	16	Esperanza de vida media al nacer, medida en años	Flotante mayor a 0
une_hiv	Flotante	3	Porcentaje de la población de adultos entre 15 y 49 años con prevalencia del VIH	Flotante entre 0 y 100
une_gni	Flotante	7	Renta Nacional Bruta (RNB) per capita, medida en dólares estadounidenses	Flotante mayor a 0
une_poverty	Flotante	3	Porcentaje de la población dentro de una taza de pobreza de 1.90 dolares al día	Flotante entre 0 y 100
une_edu_spend	Flotante	17	Gasto del gobierno en educación como porcentaje del PIB	Flotante entre 0 y 100
une_literacy	Flotante	16	Tasa de alfabetización para personas de ambos sexos mayores de 15 años	Flotante entre 0 y 100
une_school	Flotante	17	Años medios de escolaridad (ISCED mayor o igual a 1) para la población mayor de 25 años	Flotante mayor a 0
Descripción	Dataset que contiene la combinación de datos de 183 países durante los años 2000 a 2016 recopilados por el WHO y la UNESCO, con dimensión (3111, 32) y consumo de memoria de 1324636 bytes.			

2. Entendimiento de datos

2.1 Importar librerías

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
from mpl_toolkits.mplot3d import Axes3D
import plotly.express as px
from datetime import datetime
import shap
import statsmodels.api as sm
import pickle as pk
import warnings
%matplotlib inline
from scipy import stats
from scipy.stats import chi2
from scipy.stats import pearsonr
from matplotlib.pyplot import rcParams
from sklearn.cluster import DBSCAN
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression, Lasso, Ridge, ElasticNet
from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import RandomizedSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_absolute_percentage_error, mean_squared_error, mean_absolute_error
from sklearn.base import clone
from sklearn.decomposition import PCA
plt.rcParams.update({'font.size': 14})
warnings.filterwarnings('ignore')
```

2.2 Carga de datos

Se reorganizan las columnas para agrupar información similar y ubicar la variable objetivo en la última columna.

```
data_raw = data_raw.loc[:, ["region", "country", "country_code", "year", "une_pop", "gni_capita", "une_gni", "gghe-d",
                            "che_gdp", "une_edu_spend", "une_literacy", "une_school", "une_poverty", "basic_water", "doctors",
                            "hospitals", "bmi", "age5-19thinness", "age5-19obesity", "alcohol", "hepatitis", "measles", "polio",
                            "diphtheria", "une_hiv", "adult_mortality", "infant_mort", "une_infant", "age1-4mort", "life_expect",
                            "une_life", "life_exp60"]]
data_raw.head()
```

5 rows × 32 columns

	region	country	country_code	year	une_pop	gni_capita	une_gni	gghe-d	che_gdp	une_edu_spend	...	polio	diphtheria	une_hiv	adult_mortality
0	Africa	Angola	AGO	2000	16395.473	2190.0	2530.0	1.11099	1.90860	2.60753	...	21.0	31.0	1.0	383.5583
1	Africa	Angola	AGO	2001	16945.753	2290.0	2630.0	2.04631	4.48352	NaN	...	28.0	42.0	1.1	372.3876
2	Africa	Angola	AGO	2002	17519.417	2690.0	3180.0	1.30863	3.32946	NaN	...	22.0	47.0	1.2	354.5147
3	Africa	Angola	AGO	2003	18121.479	2820.0	3260.0	1.46560	3.54797	NaN	...	21.0	46.0	1.3	343.2169
4	Africa	Angola	AGO	2004	18758.145	3080.0	3560.0	1.68663	3.96720	NaN	...	18.0	47.0	1.3	333.8711

2.3 Análisis descriptivo

Ya que los datos del dataset se encuentran organizados de manera ascendente con respecto a los años, se analizan inicialmente los primeros y últimos 17 registros que corresponden a los años 2000 a 2016 de los países Angola y República Democrática Popular Lao respectivamente.

data_raw.head(17)

	region	country	country_code	year	une_pop	gini_capita	une_gni	gghe-d	che_gdp	une_edu_spend	une_literacy	une_school	une_poverty	basic_
0	Africa	Angola	AGO	2000	16395.473	2190.0	2530.0	1.11099	1.90860	2.60753	NaN	NaN	32.3	41
1	Africa	Angola	AGO	2001	16945.753	2290.0	2630.0	2.04631	4.48352	NaN	67.40542	NaN	NaN	42
2	Africa	Angola	AGO	2002	17519.417	2690.0	3180.0	1.30863	3.32946	NaN	NaN	NaN	NaN	43
3	Africa	Angola	AGO	2003	18121.479	2820.0	3260.0	1.46560	3.54797	NaN	NaN	NaN	NaN	44
4	Africa	Angola	AGO	2004	18758.145	3080.0	3560.0	1.68663	3.96720	NaN	NaN	NaN	NaN	45
5	Africa	Angola	AGO	2005	19433.602	3570.0	4060.0	1.27876	2.85220	2.12011	NaN	NaN	NaN	46
6	Africa	Angola	AGO	2006	20149.901	4260.0	4450.0	1.44412	2.68554	2.28146	NaN	NaN	NaN	47
7	Africa	Angola	AGO	2007	20905.363	5320.0	5030.0	1.72199	2.97439	NaN	NaN	NaN	NaN	47
8	Africa	Angola	AGO	2008	21695.634	5720.0	5260.0	2.13844	3.32290	NaN	NaN	NaN	30.1	48
9	Africa	Angola	AGO	2009	22514.281	6200.0	5500.0	2.60046	3.84261	NaN	NaN	NaN	NaN	49
10	Africa	Angola	AGO	2010	23356.246	6230.0	5630.0	1.67410	2.69510	3.42132	NaN	NaN	NaN	50
11	Africa	Angola	AGO	2011	24220.661	6430.0	5800.0	1.71322	2.64561	NaN	NaN	NaN	NaN	51
12	Africa	Angola	AGO	2012	25107.931	6560.0	6220.0	1.56334	2.39575	NaN	NaN	NaN	NaN	51
13	Africa	Angola	AGO	2013	26015.781	6770.0	6470.0	1.69517	2.73283	NaN	NaN	NaN	NaN	52
14	Africa	Angola	AGO	2014	26941.779	NaN	6760.0	1.31816	2.43413	NaN	66.03011	3.99596	NaN	53
15	Africa	Angola	AGO	2015	27884.381	NaN	6740.0	1.23424	2.60579	NaN	NaN	NaN	NaN	54
16	Africa	Angola	AGO	2016	28842.489	NaN	6410.0	1.19754	2.71315	NaN	NaN	NaN	NaN	55

data_raw.tail(17)

	region	country	country_code	year	une_pop	gini_capita	une_gni	gghe-d	che_gdp	une_edu_spend	une_literacy	une_school	une_poverty	basic_
3094	Western Pacific	Lao People's Democratic Republic	LAO	2000	5323.700	1770.0	1900.0	1.23306	4.27514	1.50369	69.58312	NaN	NaN	
3095	Western Pacific	Lao People's Democratic Republic	LAO	2001	5409.582	1890.0	2020.0	1.32957	4.29032	1.99028	68.73439	NaN	NaN	
3096	Western Pacific	Lao People's Democratic Republic	LAO	2002	5493.246	2010.0	2150.0	1.01875	3.69751	2.82522	NaN	NaN	33.8	
3097	Western Pacific	Lao People's Democratic Republic	LAO	2003	5576.640	2100.0	2240.0	1.09625	4.47921	NaN	NaN	NaN	NaN	
3098	Western Pacific	Lao People's Democratic Republic	LAO	2004	5662.208	2300.0	2450.0	0.79426	3.54057	2.41300	NaN	NaN	NaN	
3099	Western Pacific	Lao People's Democratic Republic	LAO	2005	5751.676	2550.0	2710.0	0.90672	3.33673	2.41409	72.70226	NaN	NaN	
3100	Western Pacific	Lao People's Democratic Republic	LAO	2006	5846.074	2710.0	2890.0	0.64222	2.93201	2.95879	NaN	NaN	NaN	
3101	Western Pacific	Lao People's Democratic Republic	LAO	2007	5944.948	3000.0	3210.0	0.57035	3.15346	3.07842	NaN	NaN	27.0	

3102	Western Pacific	Lao People's Democratic Republic	LAO	2008	6046.620	3190.0	3420.0	0.51333	2.76727	2.27855	NaN	NaN	NaN
3103	Western Pacific	Lao People's Democratic Republic	LAO	2009	6148.623	3450.0	3720.0	1.14664	3.46300	1.65485	NaN	NaN	NaN
3104	Western Pacific	Lao People's Democratic Republic	LAO	2010	6249.165	3580.0	3870.0	0.60364	2.91155	1.70979	NaN	NaN	NaN
3105	Western Pacific	Lao People's Democratic Republic	LAO	2011	6347.567	3840.0	4180.0	0.36694	1.94458	1.70802	58.28794	NaN	NaN
3106	Western Pacific	Lao People's Democratic Republic	LAO	2012	6444.530	4170.0	4570.0	0.43670	2.07638	1.82147	NaN	NaN	22.7
3107	Western Pacific	Lao People's Democratic Republic	LAO	2013	6541.304	4570.0	4980.0	0.71266	2.39985	3.23381	NaN	NaN	NaN
3108	Western Pacific	Lao People's Democratic Republic	LAO	2014	6639.756	NaN	5440.0	0.68373	2.29846	2.93781	NaN	NaN	NaN
3109	Western Pacific	Lao People's Democratic Republic	LAO	2015	6741.164	NaN	5810.0	0.86391	2.45366	NaN	84.66104	NaN	NaN
3110	Western Pacific	Lao People's Democratic Republic	LAO	2016	6845.846	NaN	6190.0	0.76495	2.36087	NaN	NaN	NaN	NaN

Empleando los métodos **head()** y **tail()**, se identifican las siguientes características de los datos contenidos en el dataset:

1. El conjunto de datos empleado cuenta con 32 columnas y 3111 registros, que corresponden a 183 países durante el período 2000-2016, y tiene un consumo de memoria total de 1324636 bytes.
2. Las variables: población (une_pop), RNB (GNI por sus siglas en inglés) per capita (gni_capita y une_gni), acceso a servicios básicos de agua potable (basic_water), BMI (bmi), porcentaje de niños y adolescentes con obesidad (age5-19obesity), población con prevalencia de VIH (une_hiv), esperanza de vida media al nacer (life_expect y une_life), y esperanza de vida media a los 60 años (life_exp60), tienden a aumentar progresivamente y con constancia a lo largo del período estudiado.
3. Las variables: Porcentaje de niños de un año que han recibido la vacuna contra la hepatitis B (hepatitis), el sarampión (measles), la polio (polio), y la difteria, tétanos y tosferina (diphtheria), y el consumo anual de alcohol por persona mayor de 15 años (alcohol), tienen una tendencia creciente fluctuante durante el período. Esto implica que, aunque la tendencia general es creciente, esta no se mantiene constante, pues hay años donde decrece con respecto al anterior.
4. Las variables: porcentaje de niños y adolescentes con delgadez (age5-19thinness), tasa de mortalidad para adultos entre 15 y 60 años (adult_mortality), tasa de mortalidad de niños menores de un año (infant_mort), número de muertes de niños menores de 1 año (une_infant), y número de muertes de niños entre 1 y 4 años (age1-4mort), tienden a decrecer progresivamente y con constancia a lo largo del período estudiado.
5. Las variables: GGHE-D (gghe-d), y CHE (che_gdp) son variables, por lo que no presentan ninguna tendencia creciente o decreciente durante el período.
6. Las variables: tasa de alfabetización (une_literacy), gasto del gobierno en educación (une_edu_spend), tasa de pobreza (une_poverty), años medios de escolaridad (une_school), número de doctores por cada 10000 habitantes (doctors), y densidad total de hospitales por cada 100000 habitantes (hospitals), presentan una gran cantidad de datos faltantes.

Aunque únicamente se logran analizar los registros de los dos países mencionados previamente, se espera que algunas de las tendencias encontradas sean comunes para la mayoría de los demás países contenidos en el dataset.

Adicionalmente, se identifican los siguientes aspectos importantes sobre el formato de los datos numéricos:

1. Los datos tipo flotante son redondeados a máximo 6 cifras decimales al pasar del archivo csv original al dataframe con el que se va a trabajar. En el archivo original, algunos de los valores flotantes tienen hasta 17 cifras decimales, sin embargo, muchas de estas cifras no son significativas, por lo que pueden ser aproximados a menos cifras decimales sin pérdida de información significativa.
2. Las variables: gni_capita, une_gni, hepatitis, measles, polio, y diphtheria, son variables enteras representadas como flotantes.
3. Las variables infant_mort y une_infant representan la mortalidad infantil, sin embargo, infant_mort está representada como una tasa por cada 1000 nacimientos vivos, mientras que une_infant representa el número de muertes por cada 1000 nacimientos vivos. Si se multiplica infant_mort por 1000 o se divide une_infant por 1000 se obtienen valores más similares.

```
data_raw.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3111 entries, 0 to 3110
Data columns (total 32 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   region           3111 non-null    object 
 1   country          3111 non-null    object 
 2   country_code     3111 non-null    object 
 3   year             3111 non-null    int64  
 4   une_pop          3074 non-null    float64
 5   gni_capita       2429 non-null    float64
 6   une_gni          2994 non-null    float64
 7   gghe-d          3011 non-null    float64
 8   che_gdp          2994 non-null    float64
 9   une_edu_spend   1825 non-null    float64
 10  une_literacy    571 non-null     float64
 11  une_school       805 non-null    float64
 12  une_poverty      913 non-null    float64
 13  basic_water     3079 non-null    float64
 14  doctors          1780 non-null    float64
 15  hospitals         130 non-null    float64
 16  bmi              3077 non-null    float64
 17  age5-19thinness 3077 non-null    float64
 18  age5-19obesity  3077 non-null    float64
 19  alcohol          3061 non-null    float64
 20  hepatitis         2542 non-null    float64
 21  measles          3092 non-null    float64
 22  polio             3092 non-null    float64
 23  diphtheria        3092 non-null    float64
 24  une_hiv           2370 non-null    float64
 25  adult_mortality  3111 non-null    float64
 26  infant_mort      3111 non-null    float64
 27  une_infant        3111 non-null    float64
 28  age1-4mort        3111 non-null    float64
 29  life_expect       3111 non-null    float64
 30  une_life          3111 non-null    float64
 31  life_exp60        3111 non-null    float64
dtypes: float64(28), int64(1), object(3)
memory usage: 777.9+ KB
```

A partir de la información general del dataset podemos determinar lo siguiente:

1. Las variables con más valores nulos en orden ascendente son: hospitals, une_literacy, une_school_une_poverty, doctors, une_edu_spend, une_hiv, gni_capita, hepatitis, une_gni, y che_gdp. Las demás variables tienen más de 3000 valores no nulos.
2. Como se mencionó previamente, las variables gni capita, une_gni, hepatitis, measles, polio, y diphtheria, son de tipo flotante cuando deberían ser enteras.
3. La variable une_pop está en una escala de miles de habitantes. Por tal motivo, acá se presenta como tipo flotante y no tipo entero.

data_raw.describe(include="all")												
	region	country	country_code	year	une_pop	gni_capita	une_gni	gghe-d	che_gdp	une_edu_spend	une_literacy	
count	3111	3111	3111	3111.000000	3.074000e+03	2429.000000	2994.000000	3011.000000	2994.000000	1825.000000	571.000000	
unique	6	183	183		NaN	NaN	NaN	NaN	NaN	NaN	NaN	
top	Europe	Angola	AGO		NaN	NaN	NaN	NaN	NaN	NaN	NaN	
freq	850	17	17		NaN	NaN	NaN	NaN	NaN	NaN	NaN	
mean	NaN	NaN	NaN	2008.000000	3.707550e+04	13397.146974	14964.832999	3.122935	6.110353	4.53293	81.984472	
std	NaN	NaN	NaN	4.899767	1.378377e+05	16258.593973	17495.137508	2.091720	2.505267	1.75428	19.665588	
min	NaN	NaN	NaN	2000.000000	7.601600e+01	250.000000	420.000000	0.062360	1.025160	0.78744	14.376040	
25%	NaN	NaN	NaN	2004.000000	2.195105e+03	2540.000000	2970.000000	1.533445	4.238798	3.26283	72.701130	
50%	NaN	NaN	NaN	2008.000000	8.544297e+03	7460.000000	8340.000000	2.601300	5.758030	4.42541	90.953740	
75%	NaN	NaN	NaN	2012.000000	2.509552e+04	18250.000000	20482.500000	4.278110	7.850327	5.49498	95.786975	
max	NaN	NaN	NaN	2016.000000	1.414049e+06	123860.000000	122670.000000	12.062730	20.413410	14.05908	99.998190	

Con base en las estadísticas descriptivas obtenidas con el método **describe()** se identifican los siguientes aspectos interesantes:

1. La región a la que pertenece la mayor cantidad de países es Europe, con registros de 50 países.
2. El país con la menor población tiene 76016 habitantes, mientras que el de mayor población tiene 1414049351 habitantes.
3. Existen diferencias significativas entre las variables gni_capita y une_gni. Ambas variables representan la Renta Nacional Bruta (RNB) per capita, medida en dólares estadounidenses. Sin embargo, gni_capita es medida por el WHO, mientras que une_gni es medida por la UNESCO. Esto demuestra que existe una diferencia en la manera que ambas organizaciones estiman este indicador.
4. La variable basic_water, que representa el porcentaje de la población usando por lo menos servicios básicos de agua potable, tiene un valor máximo de 100.000010. Un porcentaje no puede ser mayor al 100%. Por tanto, para solucionar este error, se puede aproximar el valor a 100, omitiendo el 0.00001 restante.
5. Las variables hepatitis, measles, polio, y diphtheria en ningún caso presentan un valor de 100. Esto implica que, en ningún país contenido en el dataset, el 100% de los niños menores de 1 año están vacunados contra estas enfermedades. Este valor pudo ser asignado por el WHO como un margen de error para cubrir por la porción de la población no registrada.
6. Existen diferencias entre las variables infant_mort y une_infant puestas ambas en términos del número de muertes infantiles por cada 1000 habitantes (multiplicando infant_mort por 1000). Esto se debe a que infant_mort es medida por el WHO y une_infant por la UNESCO. Así como en el caso de gni_capita y une_gni, se deduce que estas diferencias de deben a los métodos de estimación utilizados por cada organización.
7. Las variables life_expect y une_life, las cuales representan la esperanza de vida media al nacer medida en años, presentan diferencias entre ellas. De nuevo, esto se puede deber a los métodos empleados por cada organización para estimarlas.

2.4 Calidad de datos

2.4.1 Completitud

```
is_nan = data_raw.isna().sum()
is_nan
```

region	0
country	0
country_code	0
year	0
une_pop	37
gni_capita	682
une_gni	117
gghe-d	100
che_gdp	117
une_edu_spend	1286
une_literacy	2540
une_school	2306
une_poverty	2198
basic_water	32
doctors	1331
hospitals	2981
bmi	34
age5-19thinness	34
age5-19obesity	34
alcohol	50
hepatitis	569
measles	19
polio	19
diphtheria	19
une_hiv	741
adult_mortality	0
infant_mort	0
une_infant	0
age1-4mort	0
life_expect	0
une_life	0
life_exp60	0
dtype:	int64

```
total_nan = is_nan.sum()  
total_nan
```

15246

```
total_records = data_raw.shape[0]*data_raw.shape[1]  
dataset_completeness_ratio = ((total_records-total_nan)/total_records)*100  
dataset_completeness_ratio
```

84.68539054966249

El conjunto de datos cuenta con un total de **15246** datos faltantes distribuidos en los siguientes atributos:

- **hospitals:** 2981 (95.82%)
- **une_literacy:** 2540 (81.65%)
- **une_school:** 2306 (74.12%)
- **une_poverty:** 2198 (70.65%)
- **doctors:** 1331 (42.78%)
- **une_edu_spend:** 1286 (41.34%)
- **une_hiv:** 741 (23.82%)
- **gni_capita:** 682 (21.92%)
- **hepatitis:** 569 (18.29%)
- **che_gdp:** 117 (3.76%)
- **une_gni:** 117 (3.76%)
- **gghe-d:** 100 (3.21%)
- **alcohol:** 50 (1.61%)
- **une_pop:** 37 (1.19%)
- **bmi:** 34 (1.09%)
- **age5-19thinness:** 34 (1.09%)
- **age5-19obesity:** 34 (1.09%)
- **basic_water:** 32 (1.03%)
- **measles:** 19 (0.61%)
- **polio:** 19 (0.61%)
- **diphtheria:** 19 (0.61%)

Con base en lo anterior, y considerando que el conjunto de datos completo consta de 99552 entradas, el ratio de completitud del dataset es del **84.685%**.

Para dar solución a esta dimensión, se propone inicialmente eliminar por completo las variables con un porcentaje de datos faltantes mayor al 70% (hospitals, une_literacy, une_school, y une_poverty). Aunque hospitals y une_poverty pueden ser variables con una alta correlación a la esperanza de vida, la cantidad de datos faltantes es demasiado alta, resultando en probables efectos negativos sobre el modelo si se realiza la imputación de estos.

Para las variables con un porcentaje de datos faltantes menor al 50%, se realizará la imputación de estos, considerando en todo momento que los registros corresponden a un país específico en un año particular. Por tanto, para preservar las características y tendencias de cada país, la imputación seguirá una de estas dos estrategias:

- Si a un país le falta un valor en algún año, los datos se completarán con el método de interpolación de pandas por ser una serie de tiempo. El tipo de interpolación dependerá de la distribución de la variable con respecto al tiempo.
- Si a un país le faltan valores para todos los años, los datos se completarán con el promedio de la región.

```
data_raw[data_raw.isnull().sum(axis=1) > 8]["country"].value_counts()

country
South Sudan           17
Somalia               17
Democratic People's Republic of Korea 17
Sudan                  13
Montenegro            7
Afghanistan            5
Syrian Arab Republic   4
Iraq                   3
Timor-Leste             3
Libya                  2
Sao Tome and Principe  1
Yemen                  1
Albania                 1
Name: count, dtype: int64
```

Los países de South Sudan, Somalia, y Democratic People's Republic of Korea tiene datos ausentes en 8 variables para los 17 años registrados, y el país de Sudan para 13 de los 17 años.

Considerando el gran volumen de datos faltantes en el dataset, tanto para columnas como para filas, se propone eliminar las columnas hospitals, une_literacy, une_school, y une_poverty, y las filas que corresponden a los países de South Sudan, Somalia, Democratic People's Republic of Korea, y Sudan.

Por último, se identifican los países faltantes en el dataset. Entre países integrantes y países observadores, la ONU da un resultado de 195 países que existen en el mundo, sin embargo, condicionantes de tipo geopolítico hacen que Palestina no sea reconocida como estado soberano por algunos miembros de la ONU. Por tanto, la cifra para muchos queda en 194.

Según las regiones reconocidas por la OMS, esta es la distribución de países por región:

- African Region (AFRO): 47
- Region of the Americas (AMRO): 35
- Eastern Mediterranean Region (EMRO): 21
- European Region (EURO): 53
- South-East Asia Region (SEARO): 11
- Western Pacific Region (WPRO): 27

```

data_raw[data_raw["region"]=="Africa"]["country"].nunique()
47

data_raw[data_raw["region"]=="Americas"]["country"].nunique()
33

data_raw[data_raw["region"]=="Eastern Mediterranean"]["country"].nunique()
21

data_raw[data_raw["region"]=="Europe"]["country"].nunique()
50

data_raw[data_raw["region"]=="South-East Asia"]["country"].nunique()
11

data_raw[data_raw["region"]=="Western Pacific"]["country"].nunique()
21

```

Con base en lo anterior, hacen falta la siguiente cantidad de países por región:

1. African Region (AFRO): 0
2. Region of the Americas (AMRO): 2
3. Eastern Mediterranean Region (EMRO): 0
4. European Region (EURO): 3
5. South-East Asia Region (SEARO): 0
6. Western Pacific Region (WPRO): 6

Esto da un total de 11 países ausentes en el dataset. Se desconoce el motivo de esta omisión por parte del autor. No obstante, se considera que la cantidad de datos contenidos en el dataset es suficiente para construir un modelo predictivo que cumpla con los objetivos establecidos.

2.4.2 Conformidad

En los primeros y últimos 20 registros, estadísticas descriptivas, e información general del dataset no se encontraron violaciones de formato o clase. Sin embargo, si se identificaron errores de tipo de dato y valores fuera de rango.

1. **Errores de tipo de dato:** Las variables gni_capita, une_gni, hepatitis, measles, polio, y diphtheria, son de tipo flotante cuando deberían ser enteras.

```

data_raw[["gni_capita","une_gni","hepatitis","measles","polio","diphtheria"]].dtypes

gni_capita    float64
une_gni      float64
hepatitis    float64
measles      float64
polio        float64
diphtheria   float64
dtype: object

data_raw.loc[10,[ "gni_capita","une_gni","hepatitis","measles","polio","diphtheria"]]

gni_capita    6230.0
une_gni      5630.0
hepatitis     49.0
measles       62.0
polio         56.0
diphtheria   51.0
Name: 10, dtype: object

```

2. **Valores fuera de rango:** La variable basic_water tiene un valor máximo de 100.000010, excediendo el rango de 0 a 100 que delimita esta variable.

	region	country	country_code	year	une_pop	gni_capita	une_gni	gghe-d	che_gdp	une_edu_spend	une_literacy	une_school	une_poverty	basi
2295	Europe	Iceland	ISL	2000	280.435	28070.0	28670.0	7.21218	8.95113	6.44717	NaN	NaN	NaN	10
2300	Europe	Iceland	ISL	2005	294.979	33630.0	35470.0	7.44248	9.14714	7.35609	NaN	NaN	NaN	10
2372	Europe	Finland	FIN	2009	5342.262	36450.0	38320.0	6.91199	8.87985	6.48518	NaN	12.80424	NaN	10
2375	Europe	Finland	FIN	2012	5414.770	38570.0	40830.0	7.29695	9.30153	7.19254	NaN	12.74987	NaN	10

Adicionalmente, se encuentran variables con formatos que podrían ser modificados para una mayor uniformidad y facilidad de interpretación:

1. adult_mortality y une_infant representan el número de muertes, mientras que infant_mort y age1-4mort son tasas de mortalidad, cada una por cada 1000 habitantes. Para que estas cuatro variables tengan el mismo formato o escala, infant_mort y age1-4mort se pueden multiplicar por 1000 o adult_mortality y une_infant se pueden dividir por 1000.
2. une_pop representa la población de cada país en miles. Para una mayor facilidad de interpretación, se puede multiplicar por 1000.

Claramente, todas las variables que vayan a ser utilizadas para entrenar los modelos deberán ser normalizadas o estandarizadas dependiendo del caso.

Las variables categóricas region, country, y country_code no presentan errores de tipo o formato.

En el caso de region, esta solo presenta 6 datos únicos que coinciden con las 6 regiones oficiales establecidas por la OMS:

1. African Region (AFRO)
2. Region of the Americas (AMRO)
3. Eastern Mediterranean Region (EMRO)
4. European Region (EURO)
5. South-East Asia Region (SEARO)

6. Western Pacific Region (WPRO)

```
data_raw["region"].unique()  
  
array(['Africa', 'Americas', 'Eastern Mediterranean', 'Europe',  
       'South-East Asia', 'Western Pacific'], dtype=object)
```

La variable country coincide con los nombres oficiales establecidos por la OMS.

```
data_raw["country"].unique()  
  
array(['Angola', 'Burundi', 'Benin', 'Burkina Faso', 'Botswana',  
       'Central African Republic', 'United Republic of Tanzania',  
       'Uganda', 'South Africa', 'Zambia', 'Zimbabwe', 'South Sudan',  
       'Sao Tome and Principe', 'Eswatini', 'Seychelles', 'Chad', 'Togo',  
       'Namibia', 'Niger', 'Nigeria', 'Rwanda', 'Senegal', 'Sierra Leone',  
       'Madagascar', 'Mali', 'Mozambique', 'Mauritania', 'Mauritius',  
       'Malawi', 'Gambia', 'Guinea-Bissau', 'Equatorial Guinea', 'Kenya',  
       'Liberia', 'Lesotho', 'Algeria', 'Eritrea', 'Ethiopia', 'Gabon',  
       'Ghana', 'Guinea', "Côte d'Ivoire", 'Cameroon',  
       'Democratic Republic of the Congo', 'Congo', 'Comoros',  
       'Cabo Verde', 'Argentina', 'Antigua and Barbuda', 'Bahamas',  
       'Belize', 'United States of America',  
       'Saint Vincent and the Grenadines',  
       'Venezuela (Bolivarian Republic of)', 'Paraguay', 'El Salvador',  
       'Suriname', 'Trinidad and Tobago', 'Uruguay', 'Saint Lucia',  
       'Mexico', 'Nicaragua', 'Panama', 'Peru', 'Grenada', 'Guatemala',  
       'Guyana', 'Honduras', 'Haiti', 'Jamaica', 'Costa Rica', 'Cuba',  
       'Dominican Republic', 'Ecuador',  
       'Bolivia (Plurinational State of)', 'Brazil', 'Barbados', 'Canada',  
       'Chile', 'Colombia', 'Afghanistan', 'United Arab Emirates',  
       'Bahrain', 'Djibouti', 'Egypt', 'Iran (Islamic Republic of)',  
       'Somalia', 'Syrian Arab Republic', 'Tunisia', 'Yemen', 'Oman',  
       'Pakistan', 'Qatar', 'Saudi Arabia', 'Sudan', 'Iraq', 'Jordan',  
       'Kuwait', 'Lebanon', 'Libya', 'Morocco', 'Albania', 'Armenia',  
       'Austria', 'Azerbaijan', 'Belgium', 'Sweden', 'Tajikistan',  
       'Turkmenistan', 'Turkey', 'Ukraine', 'Uzbekistan', 'Serbia',  
       'Slovakia', 'Slovenia', 'Netherlands', 'Norway', 'Poland',  
       'Portugal', 'Romania', 'Russian Federation', 'Latvia',  
       'Republic of Moldova', 'Republic of North Macedonia', 'Malta',  
       'Montenegro', 'Italy', 'Kazakhstan', 'Kyrgyzstan', 'Lithuania',  
       'Luxembourg', 'Greece', 'Croatia', 'Hungary', 'Ireland', 'Iceland',  
       'Israel', 'Spain', 'Estonia', 'Finland', 'France',  
       'United Kingdom of Great Britain and Northern Ireland', 'Georgia',  
       'Czechia', 'Germany', 'Denmark', 'Bulgaria',  
       'Bosnia and Herzegovina', 'Belarus', 'Switzerland', 'Cyprus',  
       'Bangladesh', 'Bhutan', 'Indonesia', 'India', 'Sri Lanka',  
       'Maldives', 'Myanmar', 'Nepal',  
       'Democratic People's Republic of Korea', 'Thailand', 'Timor-Leste',  
       'Australia', 'Brunei Darussalam', 'China', 'Fiji',  
       'Micronesia (Federated States of)', 'Tonga', 'Viet Nam', 'Vanuatu',  
       'Samoa', 'Philippines', 'Papua New Guinea', 'Singapore',  
       'Solomon Islands', 'Mongolia', 'Malaysia', 'New Zealand', 'Japan',  
       'Cambodia', 'Kiribati', 'Republic of Korea',  
       'Lao People's Democratic Republic"], dtype=object)
```

La variable country_code coincide con el código de 3 letras establecido por la norma ISO 3166, y cada uno de estos está correctamente asignado al nombre de país correspondiente.

```

data_raw["country_code"].unique()

array(['AGO', 'BDI', 'BEN', 'BFA', 'BWA', 'CAF', 'TZA', 'UGA', 'ZAF',
       'ZMB', 'ZWE', 'SSD', 'STP', 'SWZ', 'SYC', 'TCD', 'TGO', 'NAM',
       'NER', 'NGA', 'RWA', 'SEN', 'SLE', 'MDG', 'MLI', 'MOZ', 'MRT',
       'MUS', 'MWI', 'GMB', 'GNB', 'GNQ', 'KEN', 'LBR', 'LSO', 'DZA',
       'ERI', 'ETH', 'GAB', 'GHA', 'GIN', 'CIV', 'CMR', 'COD', 'COG',
       'COM', 'CPV', 'ARG', 'ATG', 'BHS', 'BLZ', 'USA', 'VCT', 'VEN',
       'PRY', 'SLV', 'SUR', 'TTO', 'URY', 'LCA', 'MEX', 'NIC', 'PAN',
       'PER', 'GRD', 'GTM', 'GUY', 'HND', 'HTI', 'JAM', 'CRI', 'CUB',
       'DOM', 'ECU', 'BOL', 'BRA', 'BRB', 'CAN', 'CHL', 'COL', 'AFG',
       'ARE', 'BHR', 'DJI', 'EGY', 'IRN', 'SOM', 'SYR', 'TUN', 'YEM',
       'OMN', 'PAK', 'QAT', 'SAU', 'SDN', 'IRQ', 'JOR', 'KWT', 'LBN',
       'LBY', 'MAR', 'ALB', 'ARM', 'AUT', 'AZE', 'BEL', 'SWE', 'TJK',
       'TKM', 'TUR', 'UKR', 'UZB', 'SRB', 'SVK', 'SVN', 'NLD', 'NOR',
       'POL', 'PRT', 'ROU', 'RUS', 'LVA', 'MDA', 'MKD', 'MLT', 'MNE',
       'ITA', 'KAZ', 'KGZ', 'LTU', 'LUX', 'GRC', 'HRV', 'HUN', 'IRL',
       'ISL', 'ISR', 'ESP', 'EST', 'FIN', 'FRA', 'GBR', 'GEO', 'CZE',
       'DEU', 'DNK', 'BGR', 'BIH', 'BLR', 'CHE', 'CYP', 'BGD', 'BTN',
       'IDN', 'IND', 'LKA', 'MDV', 'MMR', 'NPL', 'PRK', 'THA', 'TLS',
       'AUS', 'BRN', 'CHN', 'FJI', 'FSM', 'TON', 'VNM', 'VUT', 'WSM',
       'PHL', 'PNG', 'SGP', 'SLB', 'MNG', 'MYS', 'NZL', 'JPN', 'KHM',
       'KIR', 'KOR', 'LAO'], dtype=object)

```

Tanto country como country_code tienen 183 valores únicos, lo que confirma la correspondencia entre ambas variables.

	country	country_code
count	3111	3111
unique	183	183
top	Angola	AGO
freq	17	17

2.4.3 Consistencia

Basados en la información previamente analizada, se determina que no existen posibilidades de datos contradictorios entre las variables del dataset. No obstante, los siguientes dos casos también se pueden considerar problemas de consistencia, aunque son analizados en las dimensiones de conformidad y precisión respectivamente:

1. Variable basic_water con un valor máximo de 100.000010.
2. Los pares de variables life_expect y une_life, gni_capita y une_gni, e infant_mort y une_infant, representan lo mismo, pero sus valores difieren. Esto se debe a que las primeras son medidas por el WHO y las segundas por la UNESCO.

2.4.4 Duplicidad

```
duplicates = pd.DataFrame(data_raw.duplicated(subset=["country", "year"]る,columns=['Duplicated'])
duplicate_rows = duplicates[duplicates["Duplicated"]==True]
duplicate_rows.count()
```

```
Duplicated    0
dtype: int64
```

No existen países para los cuales se hayan registrado dos o más veces el mismo año.

```
duplicates = pd.DataFrame(data_raw.duplicated(),columns=['Duplicated'])
duplicate_rows = duplicates[duplicates["Duplicated"]==True]
duplicate_rows.count()
```

```
Duplicated    0
dtype: int64
```

No existen filas completas duplicadas en el conjunto de datos.

2.4.5 Integridad

Ya que el dataset empleado fue construido por un usuario de Kaggle, la integridad es una dimensión difícil de evaluar. Los datos que este contiene provienen de la combinación de diferentes repositorios de uso público del WHO y la UNESCO.

Aunque los datos originales no sufren de problemas de integridad, pues provienen de fuentes confiables, los diferentes procedimientos realizados para crear el dataset pudieron afectar esta dimensión de calidad. Ya que los datos tuvieron que ser recopilados y posteriormente combinados, errores de transferencia de la información o errores humanos pudieron haber influido en la integridad del dataset. Esto se ve reflejado en datos borrados o agregados sin seguir el protocolo apropiado, errores en el cambio de formato al transferir una base de datos a otra, entre otros.

Considerando lo anterior, los juicios referentes a esta dimensión no tienen el suficiente respaldo. Si se quisiera determinar si los datos son íntegros, habría que hacer una validación cruzada entre el dataset y sus diferentes orígenes. Debido a la complejidad de esta tarea, se asume que los procedimientos llevados a cabo fueron realizados responsablemente, preservando la integridad de los datos en cada instancia.

Por otra parte, con referencia a la relevancia temporal de los datos, estos datan del período 2000 a 2016. Por tanto, aunque los datos no son recientes, estos reflejan adecuadamente la tendencia del aumento de la esperanza de vida que se ha presenciado en las últimas décadas. Como ejercicio académico, este dataset cumple con las características necesarias para construir un modelo predictivo relevante.

2.4.6 Exactitud

La evaluación de esta dimensión se basará en el supuesto de que todos los datos contenidos en el dataset son verídicos y representan a cabalidad la realidad. Este supuesto tiene su justificación en el propio origen de los datos. Tanto la OMS como la UNESCO son organizaciones reconocidas a nivel mundial, cuya autoridad en la recolección de datos demográficos y socioeconómicos es indiscutible. En este orden de ideas, se atribuye el error en el valor máximo de la variable basic_water a la migración de los datos llevada a cabo por el autor del dataset, y no a la OMS.

Con esto en mente, surge la problemática de los pares de variables life_expect y une_life, gni_capita y une_gni, e infant_mort y une_infant, cuyas diferencias se asume son consecuencia de los métodos de estimación de cada organización. Sin embargo, bajo el supuesto de que todos los datos representan adecuadamente la realidad, se

toma la libertad de eliminar indiscriminadamente una de las dos variables de cada pareja. Las variables que serán eliminadas son une_life, une_infant, y gni_capita.

Para justificar en cierta medida la eliminación de une_life y une_infant, asumimos que la Organización Mundial de la Salud es el ente encargado y más cualificado para estimar estos indicadores. En el caso de gni_capita, esta fue eliminada por ser la que más valores faltantes tenía.

Como parte del análisis de la exactitud de los datos, se evaluarán los outliers presentes en el dataset con la distancia de Mahalanobis y el método DBSCAN clustering. Aparte del outlier correspondiente a basic_water, según lo analizado en las estadísticas descriptivas, y basados en el supuesto de que los datos son verídicos, los demás outliers que se encuentren serán límites. Ósea, valores que se escapan del “grupo medio”, pero que serán conservados sin tratamiento alguno, pues estos representan características importantes de los países, y son de gran valor para el modelo predictivo.

No obstante, para poder realizar este análisis, se necesita hacer primero un preprocesamiento de los datos donde se imputen los datos faltantes, se eliminen las filas y columnas que no serán empleadas, y se haga la corrección del valor máximo de basic_water. Una vez hecho esto, se podrán calcular los outliers con los métodos previamente enunciados, y estos tendrán mayor validez.

3. Preparación de datos

3.1 Preprocesamiento de datos

3.1.1 Correcciones

Inicialmente se corrige el error del valor máximo de basic_water.

```
data_filtered = data_raw.copy()

data_filtered.loc[data_filtered["basic_water"]>100, "basic_water"] = 100

data_filtered[data_filtered["basic_water"]>100]["basic_water"]

Series([], Name: basic_water, dtype: float64)

data_filtered.shape

(3111, 32)
```

3.1.2 Selección de filas y columnas

A continuación, se crea el dataset mencionado en la dimensión de completitud.

Asimismo, se elimina la variable life_expect60, pues esta no es relevante para el proyecto. Ya que life_expect y life_exp60 son indicadores de esperanza de vida, la correlación entre ambas va a ser significativa. La esperanza de vida al nacer incorpora las tasas de mortalidad en todas las edades, incluyendo las etapas tempranas y avanzadas de la vida. Por tanto, hay una interrelación inherente entre la longevidad general y la longevidad específica a partir de los 60 años.

Vistas ambas variables como vectores, se podría conjeturar que life_exp60 es dependiente linealmente de life_expect. Esto se puede interpretar como que life_exp60 no aporta información nueva que pueda ser valiosa para la capacidad de generalización del modelo predictivo.

En otras palabras, como consecuencia de esta dependencia, si se utiliza esta variable para entrenar el modelo, este le otorgará una gran importancia, que opacará la influencia de las demás variables sobre la predicción. Considerando el propósito de este proyecto, no resulta coherente incluir esta variable para obtener las predicciones y las correspondientes recomendaciones basadas en la importancia de las variables. Una recomendación para aumentar la esperanza de vida media al nacer no puede ser incrementar la esperanza de vida media a los 60 años. Esto no tendría valor alguno para un gobierno que quiera implementar medidas con este objetivo, pues no podría derivar acciones concretas y áreas específicas donde invertir más recursos.

```
data_selected = data_filtered.drop(["une_life", "une_infant", "gni_capita", "hospitals", "une_literacy",  
                                    "une_school", "une_poverty", "life_exp60"], axis=1)  
data_selected.shape  
(3111, 24)
```

```
countries_to_remove = ["South Sudan", "Somalia", "Democratic People's Republic of Korea", "Sudan"]  
data_selected = data_selected.loc[~data_selected['country'].isin(countries_to_remove)]  
data_selected.shape  
(3043, 24)
```

Bajo el mismo razonamiento, se eliminan las variables infant_mort, adult_mortality, y age1-4mort.

La definición de esperanza de vida al nacer, medida en años, según el WHO es: "El número promedio de años que un recién nacido podría esperar vivir, si pasara por la vida expuesto a las tasas de mortalidad específicas por sexo y edad que prevalecen en el momento de su nacimiento, durante un año específico, en un país, territorio o área geográfica determinados."

Este indicador es derivado de tablas de mortalidad y se basa en tasas de mortalidad específicas por sexo y edad. Lo que, en otras palabras, indica que la esperanza de vida al nacer refleja el nivel de mortalidad general de una población.

En este orden de ideas, como fue el caso de life_exp60, las variables mencionadas también tienen una gran correlación con life_expect. Siguiendo el mismo argumento anterior, no es coherente incluir estas variables en el proyecto, por lo que son eliminadas del dataset.

```
data_selected = data_selected.drop(["infant_mort", "adult_mortality", "age1-4mort"], axis=1)
data_selected.shape
```

(3043, 21)

```
data_selected.tail()
```

	region	country	country_code	year	une_pop	une_gni	gghe-d	che_gdp	une_edu_spend	basic_water	doctors	bmi	age5-19thinness	age5-19obesity
3106	Western Pacific	Lao People's Democratic Republic	LAO	2012	6444.530	4570.0	0.43670	2.07638	1.82147	72.07056	1.800	22.4	9.2	3.1
3107	Western Pacific	Lao People's Democratic Republic	LAO	2013	6541.304	4980.0	0.71266	2.39985	3.23381	74.05544	4.493	22.6	9.1	3.5
3108	Western Pacific	Lao People's Democratic Republic	LAO	2014	6639.756	5440.0	0.68373	2.29846	2.93781	76.02924	4.949	22.7	9.0	3.8
3109	Western Pacific	Lao People's Democratic Republic	LAO	2015	6741.164	5810.0	0.86391	2.45366	NaN	77.99142	NaN	22.8	8.9	4.2
3110	Western Pacific	Lao People's Democratic Republic	LAO	2016	6845.846	6190.0	0.76495	2.36087	NaN	79.94190	NaN	22.9	8.9	4.7

Se resetean los índices del dataframe.

```
data_selected = data_selected.reset_index(drop=True)
data_selected.tail()
```

	region	country	country_code	year	une_pop	une_gni	gghe-d	che_gdp	une_edu_spend	basic_water	doctors	bmi	age5-19thinness	age5-19obesity
3038	Western Pacific	Lao People's Democratic Republic	LAO	2012	6444.530	4570.0	0.43670	2.07638	1.82147	72.07056	1.800	22.4	9.2	3.1
3039	Western Pacific	Lao People's Democratic Republic	LAO	2013	6541.304	4980.0	0.71266	2.39985	3.23381	74.05544	4.493	22.6	9.1	3.5
3040	Western Pacific	Lao People's Democratic Republic	LAO	2014	6639.756	5440.0	0.68373	2.29846	2.93781	76.02924	4.949	22.7	9.0	3.8
3041	Western Pacific	Lao People's Democratic Republic	LAO	2015	6741.164	5810.0	0.86391	2.45366	NaN	77.99142	NaN	22.8	8.9	4.2
3042	Western Pacific	Lao People's Democratic Republic	LAO	2016	6845.846	6190.0	0.76495	2.36087	NaN	79.94190	NaN	22.9	8.9	4.7

3.1.3 Imputación

Ahora se procede a imputar los valores faltantes.

```
region          0
country         0
country_code    0
year            0
une_pop         37
une_gni         72
gghe-d          49
che_gdp         66
une_edu_spend   1230
basic_water     21
doctors         1276
bmi              0
age5-19thinness  0
age5-19obesity   0
alcohol          22
hepatitis        533
measles          8
polio             8
diphtheria       8
une_hiv          724
life_expect      0
dtype: int64
```

Se obtiene la lista de variables que contienen datos faltantes.

```
Index(['une_pop', 'une_gni', 'gghe-d', 'che_gdp', 'une_edu_spend',
       'basic_water', 'doctors', 'alcohol', 'hepatitis', 'measles', 'polio',
       'diphtheria', 'une_hiv'],
      dtype='object')
```

Se identifican cuantos países registran únicamente valores faltantes para las variables enlistadas previamente.

Column	Num Countries
une_pop	0
une_gni	3
gghe-d	1
che_gdp	2
une_edu_spend	12
basic_water	0
doctors	0
alcohol	0
hepatitis	9
measles	0
polio	0
diphtheria	0
une_hiv	42

Se puede apreciar que para las variables une_gni, gghe-d, che_gdp, une_edu_spend, hepatitis, y une_hiv, existen países a los cuales les falta la totalidad de los valores. Esto puede tener implicaciones importantes en la imputación de los datos, pues al emplear el método de interpolación, la ausencia total de datos imposibilitará la interpolación. Para dar solución a esto, se aplicará la segunda estrategia mencionada en la dimensión de completitud de datos, la cual consiste en completar los datos faltantes con el promedio de la región.

Igualmente, se determina cuantos países presentan datos faltantes para los años 2000 y 2016 (primer y último dato respectivamente), exceptuando aquellos que presentan solo valores faltantes para toda la columna.

	Falta dato año 2000	Porcentaje 2000	Falta dato año 2016	Porcentaje 2016
une_pop	2	1.12	2	1.12
une_gni	4	2.23	2	1.12
gghe-d	5	2.79	3	1.68
che_gdp	5	2.79	3	1.68
une_edu_spend	55	30.73	67	37.43
basic_water	6	3.35	0	0.00
doctors	94	52.51	70	39.11
alcohol	4	2.23	0	0.00
hepatitis	86	48.04	0	0.00
measles	2	1.12	0	0.00
polio	2	1.12	0	0.00
diphtheria	2	1.12	0	0.00
une_hiv	1	0.56	0	0.00

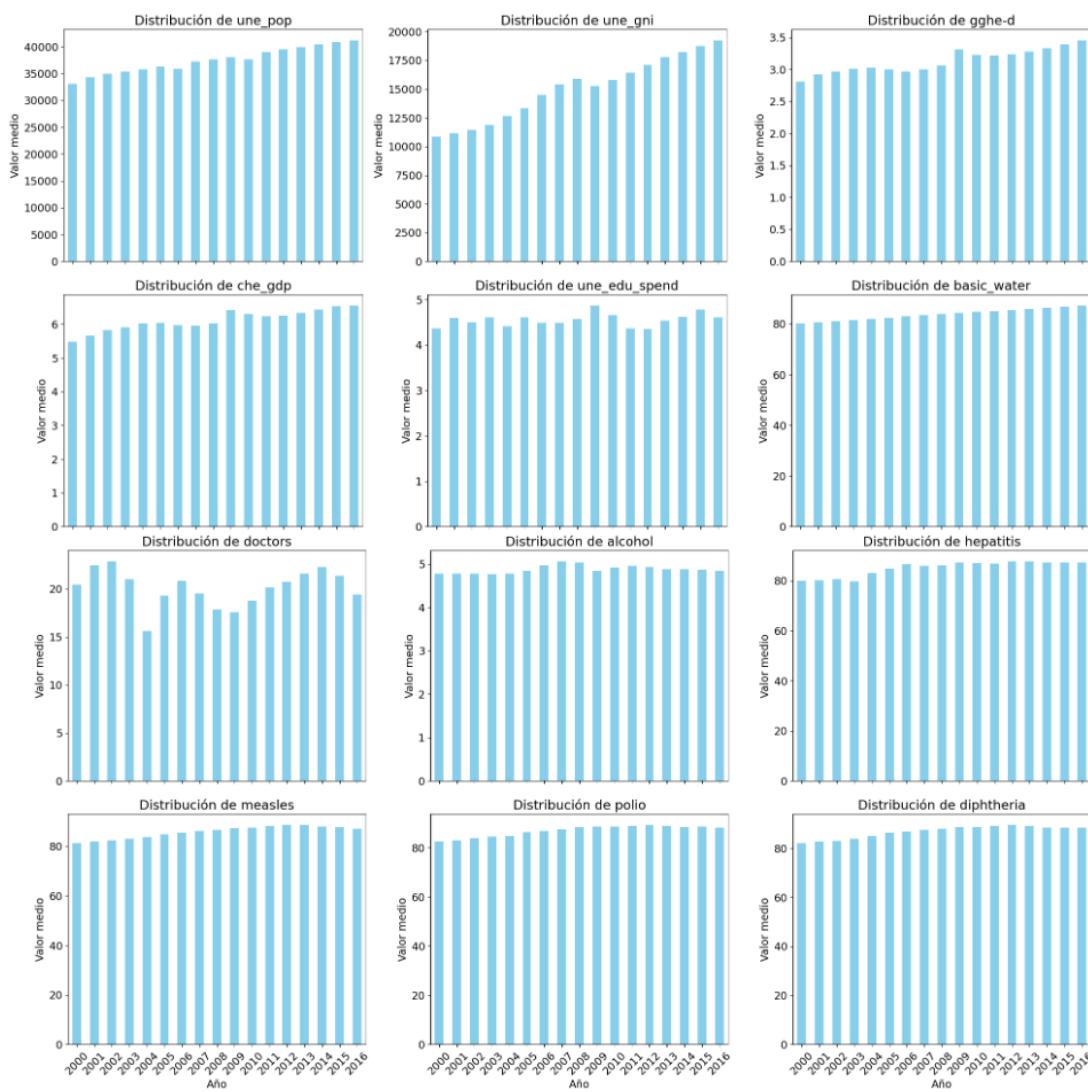
Con base en esto, podemos reconocer que las variables une_edu_spend, doctors, y hepatitis, son las que más presentan países donde el primer o último dato están ausentes. Esto se verá reflejado en los datos imputados por interpolación, puesto que, al no existir un primer o último dato, se va a presentar por lo menos un caso de NOCB (Next Observation Carried Backward) o de LOCF (Last Observation Carried Forward) en las columnas enunciadas en la tabla.

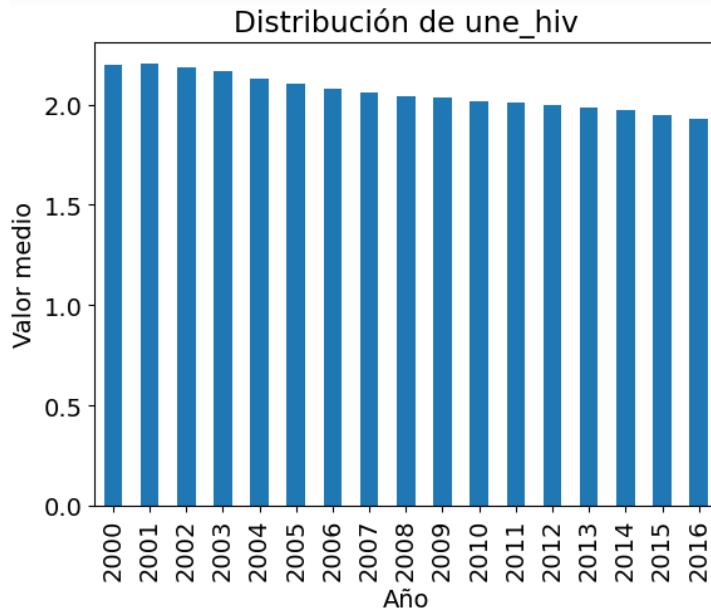
En el caso de que falte el dato para el año 2000, el interpolador completará el valor faltante con la primera siguiente observación que encuentre, lo que es equivalente a NOCB. Por otra parte, si el dato faltante es del año 2016, el interpolador le asignará la primera anterior observación encontrada, equivalente a LOCF.

El implementar estos métodos en series de tiempo donde no se cumpla la suposición de que los valores adyacentes tienden a ser iguales, ósea, datasets con poca variabilidad, puede tener consecuencias negativas. Una de ellas es la introducción de un sesgo en los datos y, debido a que este método no considera la variabilidad alrededor del valor faltante, puede conducir a una sobreestimación o subestimación de los parámetros del modelo.

Con esto en mente, aunque estos métodos de imputación no son adecuados para el presente dataset, la incapacidad de obtener los datos faltantes directamente de las organizaciones nos obliga a implementarlos.

Se procede entonces a identificar la distribución para cada variable con datos faltantes con base a su promedio anual.





A partir de las distribuciones observadas se establece que las columnas une_pop, une_gni, gghe-d, che_gdp, basic_water, alcohol, hepatitis, measles, polio, diphtheria, y une_hiv, tienen una distribución similar a la lineal. Aunque no son completamente lineales, presentando algunas variaciones a lo largo del período, se espera que el método de interpolación lineal capture la tendencia global de los datos.

En el caso de las columnas doctors y une_edu_spend, estas siguen una distribución más compleja. En el contexto de este dataset, se puede decir que las variables no siguieron ninguna tendencia, presentando variaciones impredecibles a lo largo del tiempo. Sin embargo, ya que estas columnas simplemente presentan gran variabilidad y no es que tengan una distribución polinomial, también serán imputadas con el método de interpolación lineal.

```

region          0
country         0
country_code    0
year            0
une_pop         0
une_gni         51
gghe-d          17
che_gdp         34
une_edu_spend  204
basic_water     0
doctors         0
bmi             0
age5-19thinness 0
age5-19obesity 0
alcohol          0
hepatitis        153
measles          0
polio            0
diphtheria      0
une_hiv          714
life_expect      0
dtype: int64

```

Como se mencionó previamente, en las variables une_gni, gghe-d, che_gdp, une_edu_spend, hepatitis, y une_hiv, existen países a los cuales les faltan todos los datos. Esto provoca que el interpolador no tenga valores para realizar la interpolación, dejando así una cantidad considerable de datos sin imputar.

data_selected[data_selected.isnull().any(axis=1)].head(20)																
	region	country	country_code	year	une_pop	une_gni	gghe-d	che_gdp	une_edu_spend	basic_water	doctors	bmi	age5-19thinness	age5-19obesity	alcohol	
0	Africa	Angola	AGO	2000	16395.473	2530.0	1.11099	1.90860	2.60753	41.14431	NaN	21.7	11.0	0.5	1.47439	
1	Africa	Angola	AGO	2001	16945.753	2630.0	2.04631	4.48352	NaN	42.25467	NaN	21.8	10.9	0.5	1.94025	
2	Africa	Angola	AGO	2002	17519.417	3180.0	1.30863	3.32946	NaN	43.37680	NaN	21.9	10.7	0.6	2.07512	
3	Africa	Angola	AGO	2003	18121.479	3260.0	1.46560	3.54797	NaN	44.36387	NaN	22.0	10.5	0.7	2.20275	
4	Africa	Angola	AGO	2004	18758.145	3560.0	1.68663	3.96720	NaN	45.35134	0.621	22.2	10.3	0.8	2.41274	
5	Africa	Angola	AGO	2005	19433.602	4060.0	1.27876	2.85220	2.12011	46.33602	NaN	22.3	10.2	0.8	3.48640	
6	Africa	Angola	AGO	2006	20149.901	4450.0	1.44412	2.68554	2.28146	47.15023	NaN	22.4	10.0	0.9	4.02043	
7	Africa	Angola	AGO	2007	20905.363	5030.0	1.72199	2.97439	NaN	47.96147	NaN	22.5	9.8	1.0	4.67969	
8	Africa	Angola	AGO	2008	21695.634	5260.0	2.13844	3.32290	NaN	48.77040	NaN	22.6	9.6	1.2	5.19452	
9	Africa	Angola	AGO	2009	22514.281	5500.0	2.60046	3.84261	NaN	49.57516	1.313	22.7	9.4	1.3	6.11778	
10	Africa	Angola	AGO	2010	23356.246	5630.0	1.67410	2.69510	3.42132	50.37684	NaN	22.8	9.2	1.4	6.82834	
11	Africa	Angola	AGO	2011	24220.661	5800.0	1.71322	2.64561	NaN	51.17484	NaN	22.9	9.1	1.5	6.95248	
12	Africa	Angola	AGO	2012	25107.931	6220.0	1.56334	2.39575	NaN	51.96854	NaN	23.0	8.9	1.7	7.14191	
13	Africa	Angola	AGO	2013	26015.781	6470.0	1.69517	2.73283	NaN	52.75735	NaN	23.1	8.7	1.9	7.01436	
14	Africa	Angola	AGO	2014	26941.779	6760.0	1.31816	2.43413	NaN	53.54151	NaN	23.2	8.6	2.0	7.48675	
15	Africa	Angola	AGO	2015	27884.381	6740.0	1.23424	2.60579	NaN	54.31693	NaN	23.2	8.4	2.2	5.93565	
16	Africa	Angola	AGO	2016	28842.489	6410.0	1.19754	2.71315	NaN	55.08428	NaN	23.3	8.3	2.4	5.38006	
17	Africa	Burundi	BDI	2000	6378.871	580.0	1.46507	6.17697	2.64548	50.66312	NaN	20.6	9.1	0.3	7.17131	
18	Africa	Burundi	BDI	2001	6525.545	570.0	1.69244	6.40484	2.90391	51.23447	NaN	20.7	8.9	0.4	6.62197	
19	Africa	Burundi	BDI	2002	6704.113	590.0	1.53550	6.47216	3.00493	51.80933	NaN	20.8	8.8	0.4	6.51436	

data_imputed.head(20)																
	region	country	country_code	year	une_pop	une_gni	gghe-d	che_gdp	une_edu_spend	basic_water	doctors	bmi	age5-19thinness	age5-19obesity	alcohol	
0	Africa	Angola	AGO	2000	16395.473	2530.0	1.11099	1.90860	2.607530	41.14431	0.6210	21.7	11.0	0.5	1.47439	
1	Africa	Angola	AGO	2001	16945.753	2630.0	2.04631	4.48352	2.510046	42.25467	0.6210	21.8	10.9	0.5	1.94025	
2	Africa	Angola	AGO	2002	17519.417	3180.0	1.30863	3.32946	2.412562	43.37680	0.6210	21.9	10.7	0.6	2.07512	
3	Africa	Angola	AGO	2003	18121.479	3260.0	1.46560	3.54797	2.315078	44.36387	0.6210	22.0	10.5	0.7	2.20275	
4	Africa	Angola	AGO	2004	18758.145	3560.0	1.68663	3.96720	2.217594	45.35134	0.6210	22.2	10.3	0.8	2.41274	
5	Africa	Angola	AGO	2005	19433.602	4060.0	1.27876	2.85220	2.120110	46.33602	0.7594	22.3	10.2	0.8	3.48640	
6	Africa	Angola	AGO	2006	20149.901	4450.0	1.44412	2.68554	2.281460	47.15023	0.8978	22.4	10.0	0.9	4.02043	
7	Africa	Angola	AGO	2007	20905.363	5030.0	1.72199	2.97439	2.566425	47.96147	1.0362	22.5	9.8	1.0	4.67969	
8	Africa	Angola	AGO	2008	21695.634	5260.0	2.13844	3.32290	2.851390	48.77040	1.1746	22.6	9.6	1.2	5.19452	
9	Africa	Angola	AGO	2009	22514.281	5500.0	2.60046	3.84261	3.136355	49.57516	1.3130	22.7	9.4	1.3	6.11778	
10	Africa	Angola	AGO	2010	23356.246	5630.0	1.67410	2.69510	3.421320	50.37684	1.3130	22.8	9.2	1.4	6.82834	
11	Africa	Angola	AGO	2011	24220.661	5800.0	1.71322	2.64561	3.421320	51.17484	1.3130	22.9	9.1	1.5	6.95248	
12	Africa	Angola	AGO	2012	25107.931	6220.0	1.56334	2.39575	3.421320	51.96854	1.3130	23.0	8.9	1.7	7.14191	
13	Africa	Angola	AGO	2013	26015.781	6470.0	1.69517	2.73283	3.421320	52.75735	1.3130	23.1	8.7	1.9	7.01436	
14	Africa	Angola	AGO	2014	26941.779	6760.0	1.31816	2.43413	3.421320	53.54151	1.3130	23.2	8.6	2.0	7.48675	
15	Africa	Angola	AGO	2015	27884.381	6740.0	1.23424	2.60579	3.421320	54.31693	1.3130	23.2	8.4	2.2	5.93565	
16	Africa	Angola	AGO	2016	28842.489	6410.0	1.19754	2.71315	3.421320	55.08428	1.3130	23.3	8.3	2.4	5.38006	
17	Africa	Burundi	BDI	2000	6378.871	580.0	1.46507	6.17697	2.645480	50.66312	0.2800	20.6	9.1	0.3	7.17131	
18	Africa	Burundi	BDI	2001	6525.545	570.0	1.69244	6.40484	2.903910	51.23447	0.2800	20.7	8.9	0.4	6.62197	
19	Africa	Burundi	BDI	2002	6704.113	590.0	1.53550	6.47216	3.004930	51.80933	0.2800	20.8	8.8	0.4	6.51436	

De igual forma, en el antes y el después de la imputación se puede apreciar algunos casos de NOCB y LOCF en las variables une_edu_spend, doctors, y hepatitis.

Por último, los datos faltantes restantes son imputados con el promedio de cada columna para la región a la cual pertenece el país.

```

region          0
country         0
country_code    0
year            0
une_pop         0
une_gni         0
gghe-d          0
che_gdp         0
une_edu_spend   0
basic_water     0
doctors         0
bmi             0
age5-19thinness 0
age5-19obesity  0
alcohol          0
hepatitis        0
measles          0
polio            0
diphtheria       0
une_hiv          0
life_expect      0
dtype: int64

```

3.1.4 Estandarización

En un principio, se crea un conjunto de datos únicamente con las variables numéricas.

```

df_numeric = data_imputed.copy()
df_numeric.drop(columns=["country", "country_code", "region"], inplace=True)

```

```
df_numeric.head()
```

	year	une_pop	une_gni	gghe-d	che_gdp	une_edu_spend	basic_water	doctors	bmi	age5-19thinness	age5-19obesity	alcohol	hepatitis	measles	polio
0	2000	16395.473	2530.0	1.11099	1.90860	2.607530	41.14431	0.621	21.7	11.0	0.5	1.47439	43.0	32.0	21.0
1	2001	16945.753	2630.0	2.04631	4.48352	2.510046	42.25467	0.621	21.8	10.9	0.5	1.94025	43.0	60.0	28.0
2	2002	17519.417	3180.0	1.30863	3.32946	2.412562	43.37680	0.621	21.9	10.7	0.6	2.07512	43.0	59.0	22.0
3	2003	18121.479	3260.0	1.46560	3.54797	2.315078	44.36387	0.621	22.0	10.5	0.7	2.20275	43.0	44.0	21.0
4	2004	18758.145	3560.0	1.68663	3.96720	2.217594	45.35134	0.621	22.2	10.3	0.8	2.41274	43.0	43.0	18.0

Se crean dos datasets estandarizados. El primero cuenta con todas las variables numéricas estandarizadas, incluyendo la variable objetivo life_expect. Este será empleado exclusivamente para la detección de outliers con el método de DBSCAN clustering.

	year	une_pop	une_gni	gghe-d	che_gdp	une_edu_spend	basic_water	doctors	bmi	age5-19thinness	age5-19obesity	alcohol	hepatitis
0	-1.632993	-0.150804	-0.723796	-0.969945	-1.686941	-0.986941	-2.274907	-1.061906	-1.537745	1.219074	-1.210303	-0.854576	-1.912183
1	-1.428869	-0.146834	-0.718049	-0.521602	-0.660508	-1.038075	-2.215756	-1.061906	-1.492179	1.197684	-1.210303	-0.738230	-1.912183
2	-1.224745	-0.142695	-0.686441	-0.875207	-1.120548	-1.089209	-2.155978	-1.061906	-1.446612	1.154904	-1.188289	-0.704547	-1.912183
3	-1.020621	-0.138351	-0.681843	-0.799963	-1.033444	-1.140343	-2.103394	-1.061906	-1.401046	1.112124	-1.166275	-0.672672	-1.912183
4	-0.816497	-0.133757	-0.664603	-0.694013	-0.866327	-1.191477	-2.050790	-1.061906	-1.309913	1.069345	-1.144261	-0.620227	-1.912183

El segundo dataset contiene todas las variables numéricas estandarizadas a excepción de life_expect, así como las variables categóricas region, country, y country code. Este será el conjunto de datos con el que se desarrollaran los pasos posteriores.

	region	country	country_code	year	une_pop	une_gni	gghe-d	che_gdp	une_edu_spend	basic_water	doctors	bmi	age5-19thinness
0	Africa	Angola	AGO	-1.632993	-0.150804	-0.723796	-0.969945	-1.686941	-0.986941	-2.274907	-1.061906	-1.537745	1.219074
1	Africa	Angola	AGO	-1.428869	-0.146834	-0.718049	-0.521602	-0.660508	-1.038075	-2.215756	-1.061906	-1.492179	1.197684
2	Africa	Angola	AGO	-1.224745	-0.142695	-0.686441	-0.875207	-1.120548	-1.089209	-2.155978	-1.061906	-1.446612	1.154904
3	Africa	Angola	AGO	-1.020621	-0.138351	-0.681843	-0.799963	-1.033444	-1.140343	-2.103394	-1.061906	-1.401046	1.112124
4	Africa	Angola	AGO	-0.816497	-0.133757	-0.664603	-0.694013	-0.866327	-1.191477	-2.050790	-1.061906	-1.309913	1.069345

3.1.5 Outliers

Ya realizado el preprocesamiento de los datos, en correspondencia con lo mencionado en la dimensión de exactitud de los datos, se procede a analizar los outliers presentes en el dataset imputado.

3.1.5.1 Distancia de Mahalanobis

Empleando la métrica de la distancia de Mahalanobis, calculamos los outliers presentes en el dataset de forma multivariada al usar la matriz de covarianza para determinar la distancia entre los datos y el centro de la distribución en un espacio n-dimensional.

La distancia de Mahalanobis, a diferencia de la Euclidiana, permite detectar outliers basándose en el patrón de distribución de los datos, resultando así en estimaciones más certeras.

Hay 254 outliers en el dataset para un umbral de corte de la chi-squared de 0.995

Cabe aclarar que, en este caso, la distribución chi-square es utilizada para encontrar el valor de corte, pues la distancia de Mahalanobis entrega la distancia al cuadrado (D^2). Lo que implica entonces que este umbral de corte representa aproximadamente, según la distribución chi-squared, la proporción de datos que NO son outliers. Mientras menor sea este umbral, más datos quedarán por fuera, resultando en más outliers detectados.

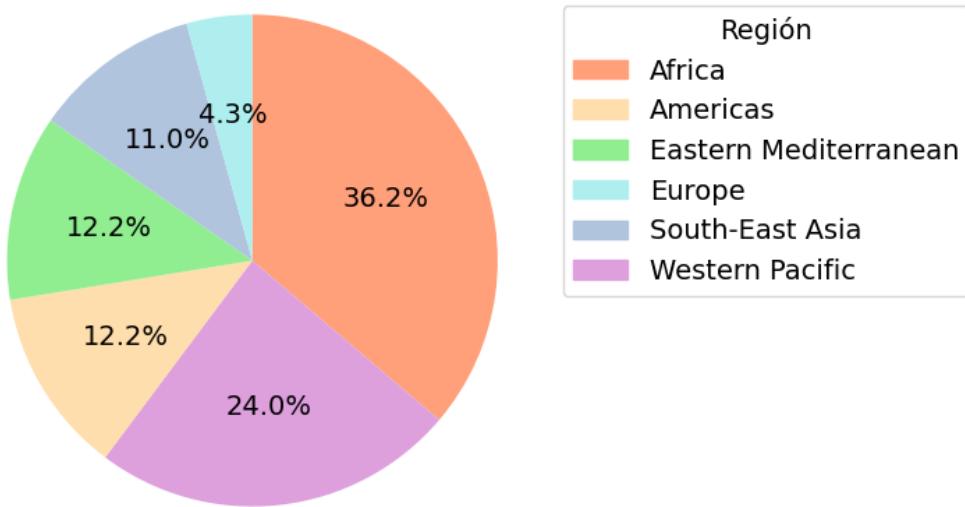
Para este análisis, se tomó un valor de corte de 0.995 y se obtuvo 254 outliers.

```
outliers["country"].unique()

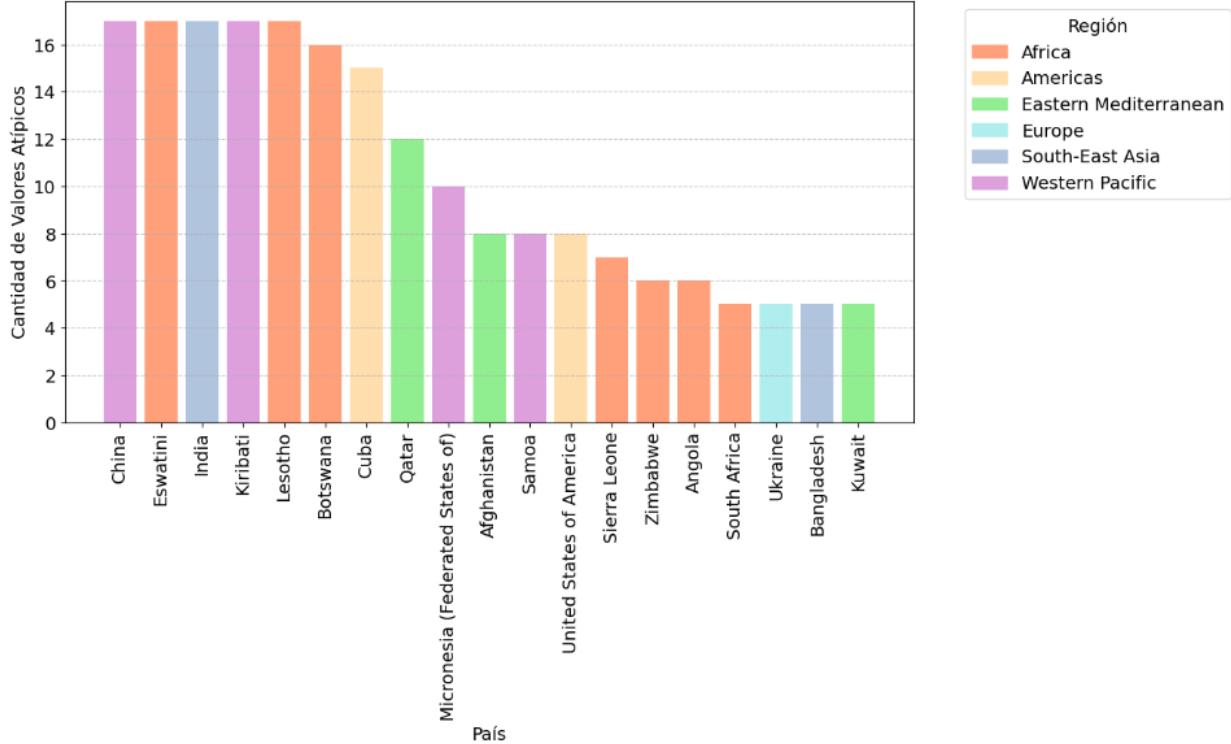
array(['Angola', 'Botswana', 'United Republic of Tanzania',
       'South Africa', 'Zimbabwe', 'Eswatini', 'Namibia', 'Niger',
       'Nigeria', 'Sierra Leone', 'Mozambique', 'Liberia', 'Lesotho',
       'Ethiopia', 'Comoros', 'Bahamas', 'United States of America',
       'Venezuela (Bolivarian Republic of)', 'Saint Lucia', 'Haiti',
       'Cuba', 'Afghanistan', 'Djibouti', 'Syrian Arab Republic', 'Yemen',
       'Qatar', 'Kuwait', 'Turkmenistan', 'Ukraine', 'Malta',
       'Montenegro', 'Georgia', 'Belarus', 'Bangladesh', 'Bhutan',
       'India', 'Myanmar', 'Timor-Leste', 'China',
       'Micronesia (Federated States of)', 'Tonga', 'Viet Nam', 'Samoa',
       'Philippines', 'Papua New Guinea', 'Singapore', 'Kiribati'],
      dtype=object)
```

Como se puede observar, los países que presentaron por lo menos 1 registro considerado outlier, son, en su mayoría, países pequeños en desarrollo. Sin embargo, países grandes desarrollados como los Estados Unidos, China e India, también presentaron outliers.

Cantidad de Outliers por Región con Distancia de Mahalanobis



Cantidad de Outliers por País (Países con 5 o más outliers) con Distancia de Mahalanobis



Con base en los gráficos, podemos identificar que la región donde se presentan más outliers es África, con un 36.2% del total. De estos outliers, Eswatini y Lesotho presentan 17 outliers cada uno. Esto implica que todos los registros desde el 2000 hasta el 2016 de estos tres países quedaron por fuera del umbral establecido. Otros países de la región de África que presentaron más de 5 outliers fueron Botswana, Sierra Leone, Zimbabwe, Angola, y South Africa.

La segunda región con más outliers es Pacífico Occidental, con 24.0%. China y Kiribati presentaron 17 outliers, seguidos por Micronesia con 10 y Samoa con 8.

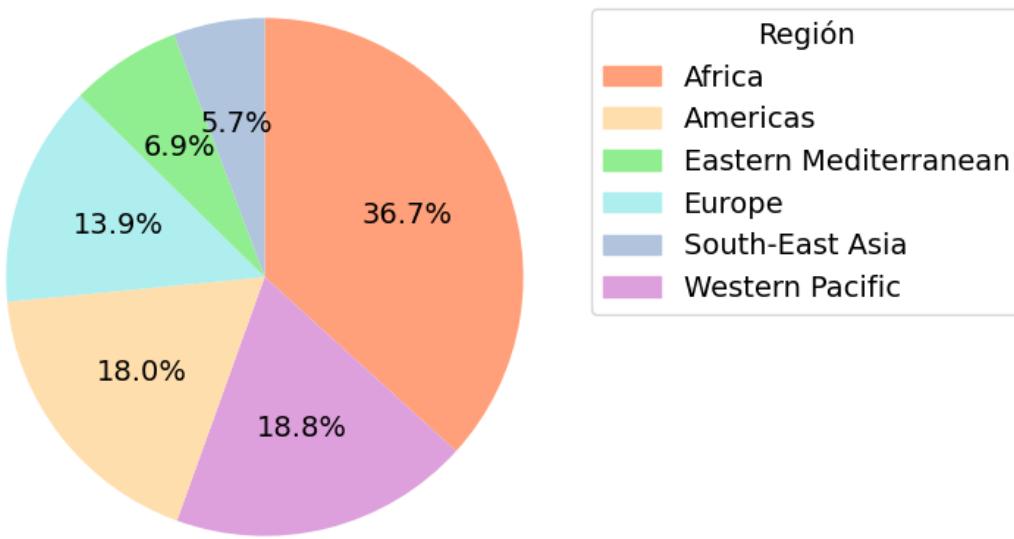
La tercera y cuarta región son las Américas y Mediterráneo Oriental, ambas con 12.2%. Para la primera, Cuba fue el país con más outliers con 15. Para la segunda, lo fue Qatar con 12.

Por último, el sudeste asiático y Europa representaron el 11.0% y 4.3% respectivamente, siendo India el país con más outliers para la primera, y Ukraine para la segunda.

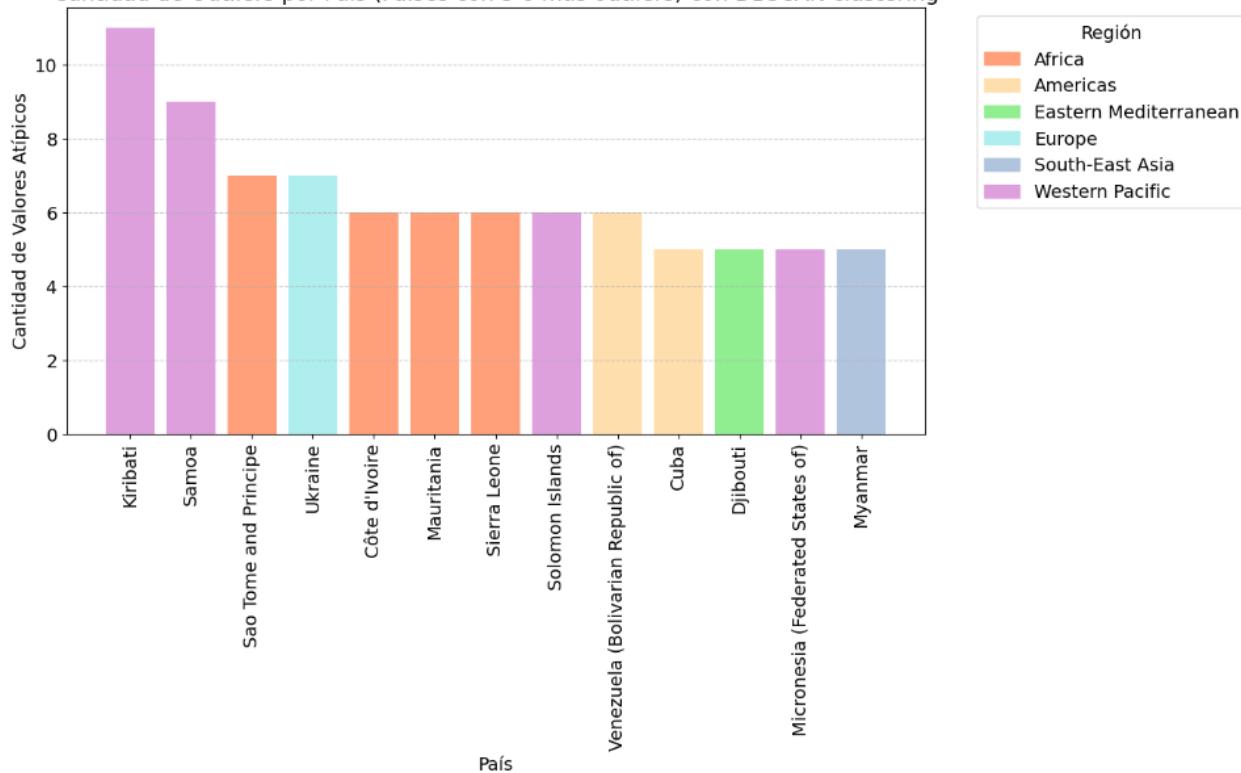
3.1.5.2 DBSCAN Clustering

Total number of outliers : 245

Cantidad de Outliers por Región con DBSCAN clustering



Cantidad de Outliers por País (Países con 5 o más outliers) con DBSCAN clustering

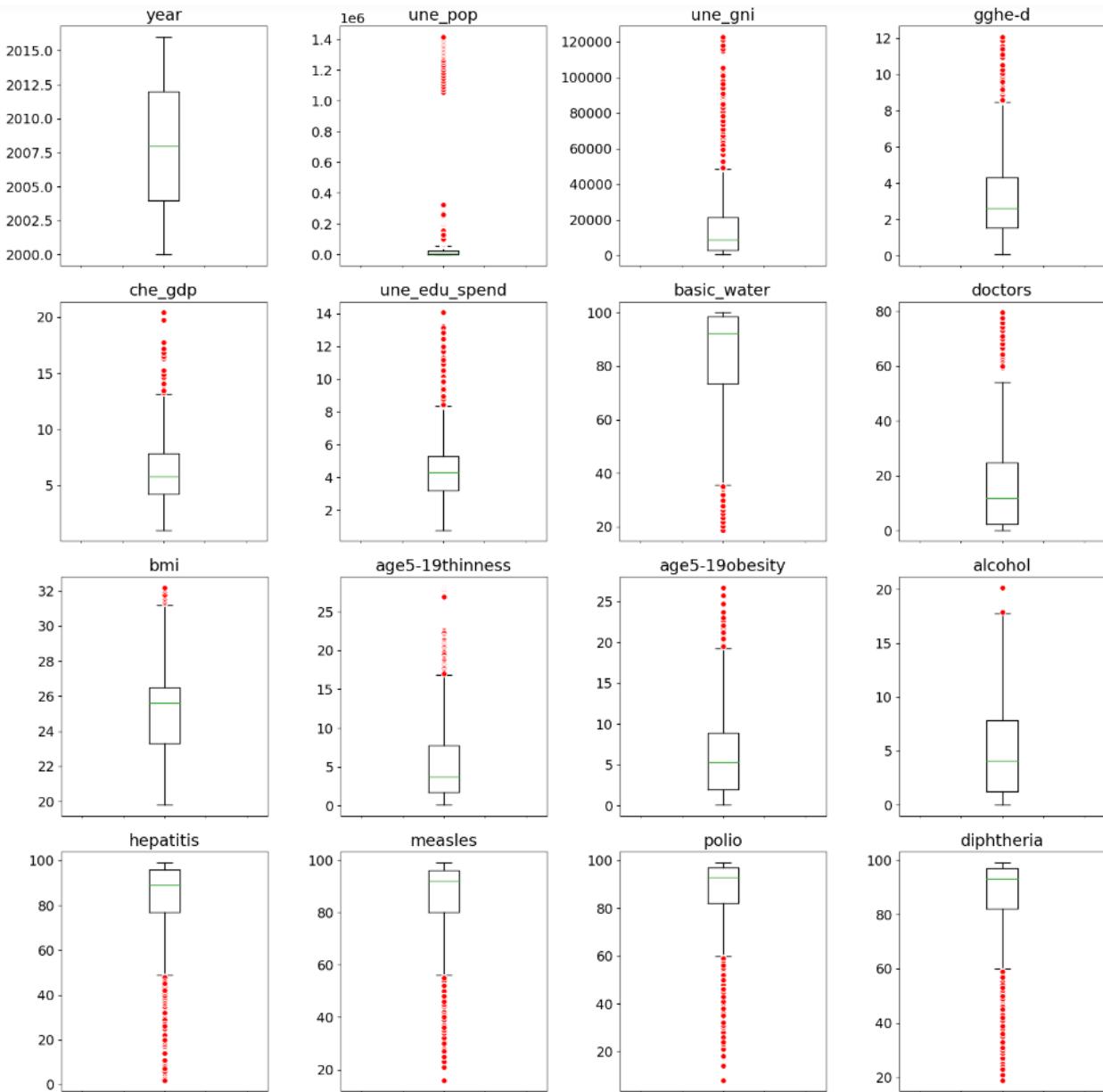


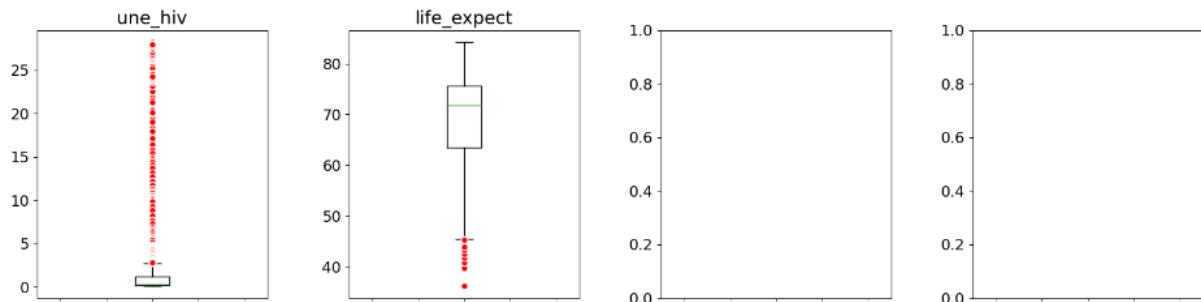
Las diferencias en los resultados obtenidos con la distancia de Mahalanobis y el DBSCAN clustering se deben al método de identificación de outliers que tiene cada uno. La distancia de Mahalanobis identifica outliers basándose en la posición relativa de los puntos en un espacio multidimensional considerando la distribución global de los datos. El DBSCAN clustering los identifica en función de la densidad local de puntos, etiquetando puntos que no están suficientemente cerca de otros como outliers. Por lo que uno reconoce puntos que son inusualmente distantes del centro de los datos, y el otro, puntos que están en regiones de baja densidad, es decir, que no tienen suficientes vecinos cercanos.

De igual forma, cada uno de ellos depende de parámetros diferentes. Mahalanobis depende principalmente de la media y la covarianza del dataset, y DBSCAN depende críticamente de los parámetros ϵ y minPts, que deben ser seleccionados adecuadamente para obtener buenos resultados.

En este orden de ideas, resulta lógico que los dos métodos identifiquen outliers diferentes, como se observó en las gráficas.

3.1.5.3 Boxplots





A partir de estos boxplots podemos reconocer la distribución de los valores atípicos para cada una de las variables. Siendo el propósito de esta sección identificar outliers, no se ahondará en la distribución de las variables ni su interpretación.

Las tendencias principales de los valores atípicos se resumen en los siguientes dos casos:

- Para las variables basic_water, hepatitis, measles, polio, diphtheria, y life_expect, los valores atípicos están por debajo del límite inferior.
- Para las variables une_pop, une_gni, gghe-d, che_gdp, une_edu_spend, doctors, bmi, age5-19thinness, age5-19obesity, alcohol, y une_hiv, los valores atípicos están por encima del límite superior.

3.2 Preparación de características

Se aplica encoding para convertir la variable region de categórica a numérica.

	country	country_code	year	une_pop	une_gni	gghe-d	che_gdp	une_edu_spend	basic_water	doctors	bmi	age5-19thinness	age5-19obesity
0	Angola	AGO	-1.632993	-0.150804	-0.723796	-0.969945	-1.686941	-0.986941	-2.274907	-1.061906	-1.537745	1.219074	-1.210303
1	Angola	AGO	-1.428869	-0.146834	-0.718049	-0.521602	-0.660508	-1.038075	-2.215756	-1.061906	-1.492179	1.197684	-1.210303
2	Angola	AGO	-1.224745	-0.142695	-0.686441	-0.875207	-1.120548	-1.089209	-2.155978	-1.061906	-1.446612	1.154904	-1.188289
3	Angola	AGO	-1.020621	-0.138351	-0.681843	-0.799963	-1.033444	-1.140343	-2.103394	-1.061906	-1.401046	1.112124	-1.166275
4	Angola	AGO	-0.816497	-0.133757	-0.664603	-0.694013	-0.866327	-1.191477	-2.050790	-1.061906	-1.309913	1.069345	-1.144261

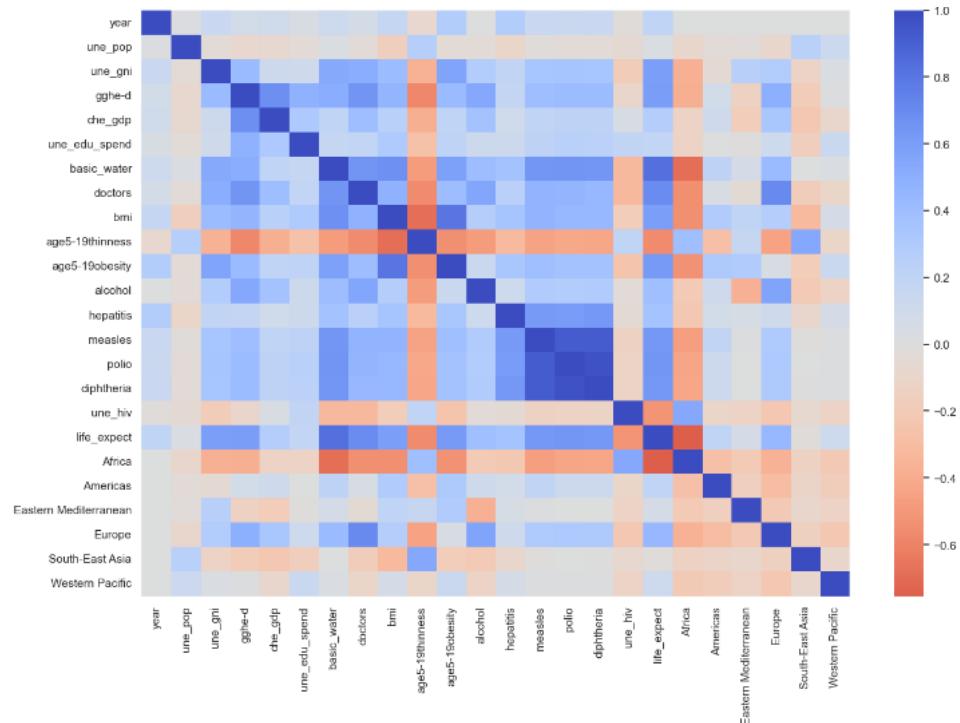
age5-thinness	age5-19obesity	alcohol	hepatitis	measles	polio	diphtheria	une_hiv	life_expect	Africa	Americas	Eastern Mediterranean	Europe	South-East Asia	Western Pacific
1.219074	-1.210303	-0.854576	-1.912183	-3.613611	-4.514793	-3.744746	-0.173087	47.33730	1	0	0	0	0	0
1.197684	-1.210303	-0.738230	-1.912183	-1.734191	-4.034952	-3.006311	-0.148883	48.19789	1	0	0	0	0	0
1.154904	-1.188289	-0.704547	-1.912183	-1.801313	-4.446244	-2.670659	-0.124680	49.42569	1	0	0	0	0	0
1.112124	-1.166275	-0.672672	-1.912183	-2.808145	-4.514793	-2.737789	-0.100476	50.50266	1	0	0	0	0	0
1.069345	-1.144261	-0.620227	-1.912183	-2.875267	-4.720438	-2.670659	-0.100476	51.52863	1	0	0	0	0	0

Las variables country y country_code son eliminadas, pues estas no serán empleadas para entrenar el modelo.

	year	une_pop	une_gni	gghe-d	che_gdp	une_edu_spend	basic_water	doctors	bmi	age5-19thinness	age5-19obesity	alcohol	hepatitis
0	-1.632993	-0.150804	-0.723796	-0.969945	-1.686941	-0.986941	-2.274907	-1.061906	-1.537745	1.219074	-1.210303	-0.854576	-1.912183
1	-1.428869	-0.146834	-0.718049	-0.521602	-0.660508	-1.038075	-2.215756	-1.061906	-1.492179	1.197684	-1.210303	-0.738230	-1.912183
2	-1.224745	-0.142695	-0.686441	-0.875207	-1.120548	-1.089209	-2.155978	-1.061906	-1.446612	1.154904	-1.188289	-0.704547	-1.912183
3	-1.020621	-0.138351	-0.681843	-0.799963	-1.033444	-1.140343	-2.103394	-1.061906	-1.401046	1.112124	-1.166275	-0.672672	-1.912183
4	-0.816497	-0.133757	-0.664603	-0.694013	-0.866327	-1.191477	-2.050790	-1.061906	-1.309913	1.069345	-1.144261	-0.620227	-1.912183
5	-0.612372	-0.128884	-0.635868	-0.889525	-1.310796	-1.242612	-1.998334	-1.051688	-1.264346	1.047955	-1.144261	-0.352085	-1.912183
6	-0.408248	-0.123716	-0.613456	-0.810260	-1.377232	-1.157977	-1.954960	-1.041470	-1.218780	1.005175	-1.122247	-0.218714	-1.912183
7	-0.204124	-0.118265	-0.580124	-0.677064	-1.262088	-1.008502	-1.911744	-1.031252	-1.173213	0.962395	-1.100232	-0.054066	-1.912183
8	0.000000	-0.112564	-0.566906	-0.477440	-1.123163	-0.859026	-1.868650	-1.021034	-1.127647	0.919615	-1.056204	0.074511	-1.813890
9	0.204124	-0.106657	-0.553113	-0.255972	-0.915992	-0.709551	-1.825779	-1.010816	-1.082080	0.876835	-1.034190	0.305091	-1.715598
10	0.408248	-0.100582	-0.545642	-0.700020	-1.373421	-0.560076	-1.783072	-1.010816	-1.036514	0.834055	-1.012176	0.482550	-1.617306

3.3 Importancia de variables

3.3.1 Correlación de Pearson



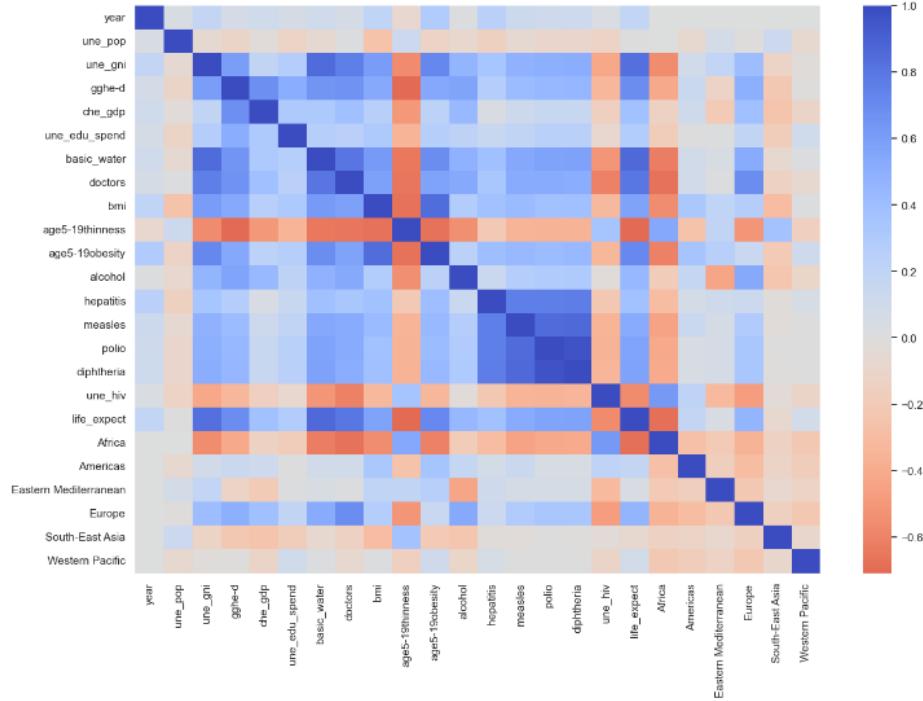
De la matriz de correlación de Pearson se identifican los siguientes aspectos con respecto a la esperanza de vida (**life_expect**):

- Hay una correlación levemente positiva con el año (**year**). Esta relación no es causal, simplemente representa la tendencia positiva de la esperanza de vida en las últimas décadas. Esta tendencia se debe a otros factores como avances en diferentes campos que indirectamente resultan en el aumento de la esperanza de vida.
- No existe una correlación significativa con la población del país (**une_pop**). Esto tiene sentido, pues una mayor o menor población no necesariamente deriva en una mayor o menor esperanza de vida.
- Hay una correlación positiva con la Renta Nacional Bruta per capita (**une_gni**). Se espera que, en general, mientras más ingresos anuales tenga un país, mayor es la proporción de recursos dirigidos a la mejora de la calidad de vida de los habitantes.

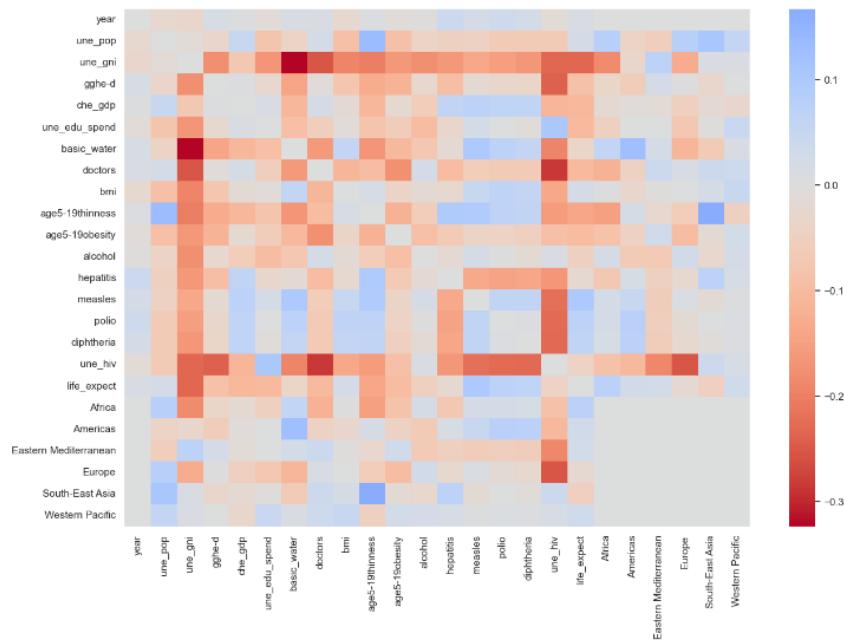
- Hay una correlación positiva con el gasto interno general del gobierno en salud (**gghe-d**), y el gasto actual en salud (**che_gdp**). El primero es un indicador del papel de las fuentes de ingresos internas generales del gobierno en la financiación de la atención sanitaria en comparación a las fuentes internas privadas y externas. El segundo indica el nivel de recursos canalizados a la salud en relación con otros usos. En otras palabras, ambas variables muestran la importancia del sector de la salud en toda la economía e indican la prioridad social que le es otorgada. Por ello, es coherente que el aumento de estos indicadores resulte en el aumento de la esperanza de vida.
- Hay una correlación positiva fuerte con el porcentaje de la población que usa por lo menos servicios básicos de agua potable (**basic_water**). Esto es lógico, pues el agua es un recurso indispensable para la vida, por tanto, el no tener acceso a agua potable imposibilita la supervivencia.
- Hay una correlación positiva medianamente fuerte con el número de doctores por cada 10000 habitantes (**doctors**). Es una relación racional, siendo evidente que mientras más disponibilidad de doctores haya, mayor es la posibilidad de que una persona reciba atención médica que pueda prolongar su vida.
- Hay una correlación positiva con el índice de masa corporal promedio de la población (**bmi**). Un argumento para dar sentido a esta relación es que, un bmi promedio mayor puede representar un mayor acceso a fuentes de alimentación que mantienen a la población por encima del rango de desnutrición y delgadez. Sin embargo, no se puede asumir que el aumento del bmi resulta en una mayor esperanza de vida. Con base en el boxplot de esta variable, se puede apreciar que el 75% de los registros para todos los países indican una población con un bmi en el umbral entre peso saludable (18.5-24.9) y sobrepeso (25-29.9). No obstante, el otro 25% se encuentra en el rango de sobrepeso y obesidad (>30). Este último rango puede representar un riesgo significativo para la salud del individuo, lo que, en consecuencia, puede reducir la esperanza de vida. Con esto en mente, es importante recordar que correlación no es lo mismo que causalidad.
- Hay una correlación negativa para los niños entre 5 y 19 años con delgadez (**age5-19thinness**) y una positiva para los niños obesos con este mismo rango de edad (**age5-19obesity**). Esta relación resulta interesante, y va de la mano con la correlación positiva entre bmi y esperanza de vida. Por un lado, la correlación negativa entre esperanza de vida y delgadez tiene sentido. No obstante, a pesar de las comprobadas consecuencias negativas que tiene la obesidad sobre la salud, ¿por qué la obesidad tiene una correlación positiva con la esperanza de vida? Para comprender esto, es importante recordar que muchas enfermedades crónicas producen una pérdida significativa de peso previo a la muerte. Esta pérdida de peso involuntaria deriva en un bmi por debajo de los niveles saludables, y, aunque la causa de muerte no es la delgadez, esta es comúnmente asociada a un mayor riesgo de mortalidad. A menos de que una persona obesa muera repentinamente, estas usualmente pierden mucho peso antes de morir por alguna enfermedad. Por tanto, la obesidad, analizada desde esta perspectiva, no se asocia con una menor esperanza de vida. Todo esto explica el por qué de las correlaciones enunciadas.
- Hay una relación positiva con el consumo anual de alcohol por persona mayor de 15 años (**alcohol**). Este es un caso similar al de **bmi** y **age5-19obesity**, donde correlación no implica causalidad. Aunque existen estudios que señalan que el consumo moderado de alcohol puede ser beneficioso para la salud, la mayoría de las fuentes científicas coinciden en que un consumo excesivo resulta en todo lo contrario. Con esto en mente, una hipótesis que puede explicar esta relación es que los países con mayor esperanza de vida consumen más alcohol. Esto se puede deber a otros factores como la calidad de vida, acceso general a bebidas alcohólicas, o capacidad adquisitiva de su población.
- Hay una relación positiva con el porcentaje de niños de 1 año que han recibido la vacuna contra la hepatitis B (**hepatitis**), el sarampión (**measles**), la polio (**polio**), y la vacuna DTP3 contra la difteria, el toxoide tetánico y la tos ferina (**diphtheria**). Es lógico que mientras mayor sea la inmunización de la población contra estas enfermedades altamente contagiosas, mayor será la inmunidad de rebaño, y menos los casos fatales.
- Hay una correlación negativa con el porcentaje de la población de adultos entre 19 y 49 años con prevalencia del VIH (**une_hiv**). Es lógico que, a menor cantidad de personas con VIH, mayor es la esperanza de vida de la población.
- Hay una correlación altamente negativa con los países de la región de África (**Africa**), levemente negativa para la región del Mediterráneo Oriental (**Eastern Mediterranean**) y Sudeste de Asia (**South-East Asia**), neutra para el Pacífico Occidental (**Western Pacific**), levemente positiva para las Américas

(Americas), y positiva para Europa (**Europe**). A partir de la percepción general que se tiene sobre la distribución de riquezas y recursos, así como la calidad de vida en diferentes regiones del mundo, estas relaciones cobran sentido. En especial, la de África y Europa.

3.3.2 Correlación de Spearman



Vista de manera superficial, se podría asumir que las matrices de correlación de Pearson y Spearman son en gran medida iguales. Sin embargo, al compararlas obtenemos algunas correlaciones iguales, pero en su mayoría, diferentes.



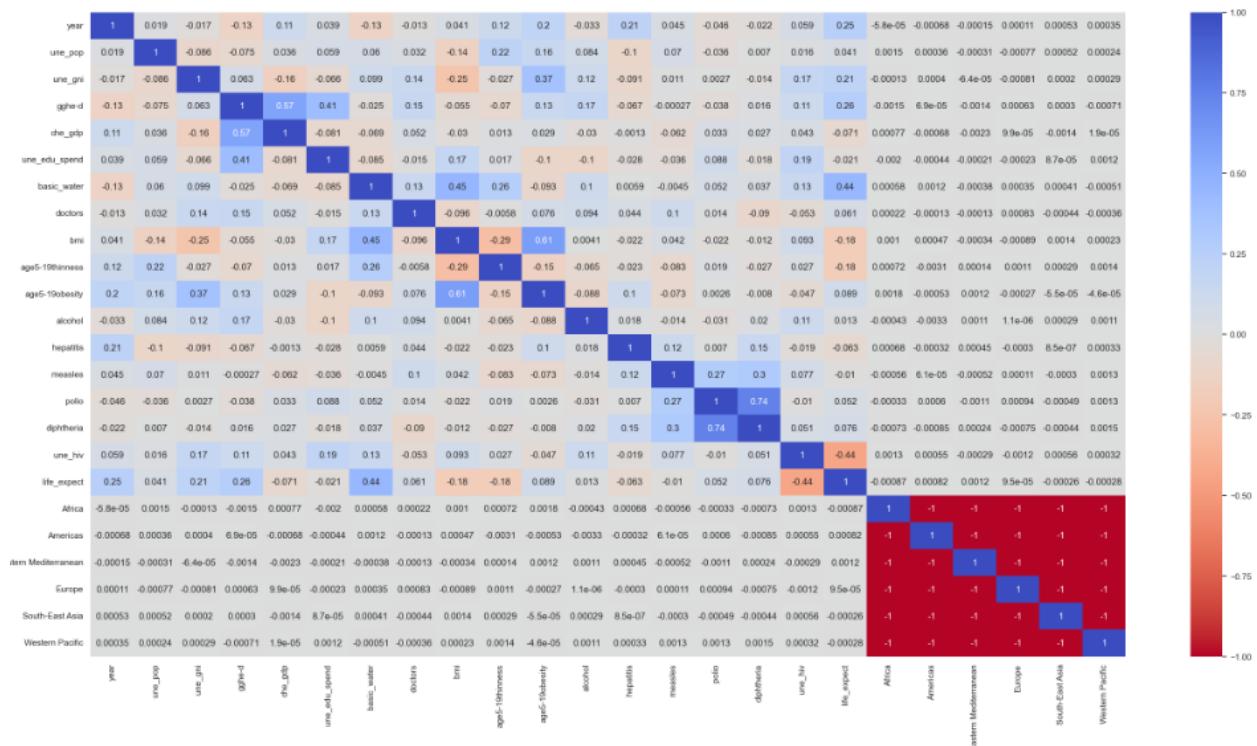
Estas diferencias entre ambas matrices se pueden deber a dos factores principales:

- Relaciones no lineales predominantes:** Las diferencias indican que la mayoría de las relaciones entre las variables no son lineales. La correlación de Spearman captura relaciones monótonas (no necesariamente lineales), mientras que Pearson se centra únicamente en relaciones lineales. Por lo tanto, las discrepancias sugieren que muchas de las relaciones en el dataset siguen patrones que no son estrictamente lineales.
- Presencia de Outliers:** La correlación de Pearson es sensible a los outliers, mientras que la correlación de Spearman, al trabajar con rangos, es más robusta frente a ellos. Por lo tanto, si hay outliers, las correlaciones de Pearson se verán más afectadas, resultando en valores diferentes a los de Spearman.

Por el contrario, las similitudes entre ambas matrices se pueden atribuir a:

- Relaciones Lineales Fuertes:** Donde las matrices son iguales, las relaciones entre las variables son probablemente tanto lineales como monótonas. Esto significa que, en esos casos específicos, las variables relacionadas siguen una tendencia lineal clara y consistente.
- Datos Sin Outliers:** Las similitudes pueden indicar que en esas áreas del dataset no hay outliers que afecten significativamente la relación entre las variables. Como se mencionó previamente, los outliers pueden tener un impacto mayor en la correlación de Pearson, pero si están ausentes, ambas correlaciones pueden ser similares.

3.3.3 Correlación parcial



La matriz de correlación parcial mide la relación entre dos variables, mientras se controla (o se elimina) el efecto de otras variables presentes en el modelo. Con esto en mente, al existir una correlación parcial de aproximadamente 0 entre las variables de región (Africa, Americas, Eastern Mediterranean, Europe, South-East Asia, y Western Pacific) y life_expect, implica que estas no tienen una relación directa con la esperanza de vida (life_expect). En otras palabras, las diferencias en la esperanza de vida no pueden ser explicadas por el hecho de que un país pertenezca a una de estas regiones, después de haber eliminado la influencia de las demás variables.

Esto sugiere entonces que las variaciones de life_expect ya están suficientemente explicadas por otras variables en el dataset. Las variables relacionadas con la región no añaden información adicional ya que estas están correlacionadas con otras variables en el dataset que explican mejor la variabilidad en la esperanza de vida. Lo más probable es que, los factores socioeconómicos y de salud que varían por región, están capturando la influencia regional de manera más efectiva.

Adicionalmente, se evidencia multicolinealidad exacta entre las variables de región. La correlación perfectamente negativa implica que se incumple el supuesto de Gauss-Markov (Paula Rodó, 2019). Según José Francisco López (2017), el Teorema de Gauss-Márkov es un conjunto de supuestos que debe cumplir un estimador MCO (Mínimo Cuadrados Ordinarios) para que se considere ELIO (Estimador lineal insesgado óptimo). Según este teorema, las variables explicativas en una muestra no pueden ser constantes y no deben existir relaciones lineales exactas entre variables explicativas (no multicolinealidad exacta).

Paula Rodó plantea, en el blog Economipedia (2019), que la multicolinealidad exacta se produce cuando más de dos variables independientes son una combinación lineal de otras variables independientes de la regresión. De no solucionarse este problema, la multicolinealidad de estas variables puede tener efectos negativos en los resultados de algunos algoritmos de machine learning.

Por este motivo, se eliminarán las variables de región, comprobando inicialmente su importancia y el efecto de su eliminación sobre los resultados del modelo en backward selection.

En referencia a otras variables cuya correlación parcial con life_expect es pequeña, como lo es el caso de measles, alcohol y une_edu_spend, estas no serán eliminadas a menos que el método de backward selection lo indique así.

3.3.4 Importancia con árboles de decisión

Para poder analizar la importancia de variables con árboles de decisión de tal forma que sea congruente con los modelos entrenados más adelante, se utilizarán los mismos conjuntos de entrenamiento y validación iniciales.

Una vez seleccionadas las variables, se crearán nuevos conjuntos de entrenamiento, validación y prueba con solo estas columnas para ser utilizados en los modelos definitivos.

3.3.4.1 División de datos en entrenamiento, validación y prueba

Se define la semilla para reproducibilidad.

semilla = 42

El enfoque del proyecto es predecir la esperanza de vida de un país con base a diversos datos proporcionados y, a partir de esto, determinar las áreas de mejora que resultarían en el aumento de la esperanza de vida. Por tanto, los modelos deben ser entrenados para identificar las correlaciones entre las variables de cada país más que las tendencias temporales.

Para lograr esto, se propone dividir el dataset en los conjuntos de entrenamiento, validación y prueba en función de los países, y no de los años, o simplemente de registros seleccionados aleatoriamente. Por ello, se selecciona aleatoriamente el 60% de los países para entrenamiento, 20% para validación, y 20% para prueba. Ya que el dataframe data_prepared no contiene la columna "country", se extraen los índices de las filas de cada país seleccionado en el dataframe data_standar. Con estos índices, se extraen las correspondientes filas de data_prepared para generar los conjuntos de datos.

```
Datos de entrenamiento: (1819,)
```

```
Datos de validación: (595,)
```

```
Datos de prueba: (629,)
```

```
Países de entrenamiento: ['Colombia' 'Namibia' 'Guyana' 'Poland' 'Canada' 'Rwanda' 'Montenegro'  
'Mozambique' 'Honduras' 'Sri Lanka' 'Cabo Verde'  
'Democratic Republic of the Congo' 'Slovenia' 'Djibouti' 'Haiti' 'Togo'  
'Norway' 'Vanuatu' 'Jamaica' 'Latvia' 'Denmark' 'Equatorial Guinea'  
'Zambia' 'Republic of Moldova' 'Nepal' 'Nigeria' 'Iceland' 'Costa Rica'  
'Lebanon' 'Italy' 'Albania' 'Papua New Guinea' 'Austria' 'Georgia'  
'Guinea-Bissau' 'Portugal' 'Australia' 'Kenya' 'Eswatini'  
'Saint Vincent and the Grenadines' 'Cameroon' 'Syrian Arab Republic'  
'Jordan' 'Trinidad and Tobago' 'Ghana' 'Bosnia and Herzegovina' 'Qatar'  
'Germany' 'Cyprus' 'Benin' 'Tajikistan' 'Luxembourg' 'Switzerland'  
'Suriname' 'Kazakhstan' 'Nicaragua' 'Mauritius' 'Myanmar' 'Libya'  
'Serbia' 'Solomon Islands' 'Ethiopia' 'Belarus' 'Morocco' 'Madagascar'  
'Malaysia' 'Bangladesh' 'Uzbekistan' 'Sao Tome and Principe'  
'United Republic of Tanzania' 'Malawi' 'China' 'Botswana' 'Liberia'  
'Barbados' 'Lithuania' 'Mongolia' 'Zimbabwe' 'Netherlands' 'Greece'  
'Angola' 'Estonia' 'Cuba' 'Guatemala' 'Comoros' 'Azerbaijan' 'Gambia'  
'Côte d'Ivoire' 'France'  
'United Kingdom of Great Britain and Northern Ireland' 'Mauritania'  
'Mali' 'Israel' 'Bahrain' 'Afghanistan' 'Iran (Islamic Republic of)'  
'Guinea' 'Cambodia' 'Tunisia' 'Kyrgyzstan' 'Japan' 'Indonesia'  
'Antigua and Barbuda' 'Kuwait' 'Finland' 'Philippines' 'Panama']
```

```
Países de validación: ['Malta' 'Bhutan' 'Iraq' 'Turkey' 'Central African Republic'  
'Bolivia (Plurinational State of)' 'Ecuador' 'Burundi'  
'Brunei Darussalam' 'Ireland' 'Timor-Leste' 'Egypt'  
'Micronesia (Federated States of)' 'Seychelles' 'Uruguay' 'Ukraine'  
'Croatia' 'Chile' 'India' 'Fiji' 'Pakistan' 'Dominican Republic'  
'Belgium' 'Uganda' 'Lesotho' 'Romania' 'Viet Nam' 'Spain' 'Bulgaria'  
'Niger' 'Bahamas' 'Paraguay' 'Maldives' 'Belize' 'South Africa']
```

```
Países de prueba: ['Algeria' 'Argentina' 'Armenia' 'Brazil' 'Burkina Faso' 'Chad' 'Congo'  
'Czechia' 'El Salvador' 'Eritrea' 'Gabon' 'Grenada' 'Hungary' 'Kiribati'  
'Lao People's Democratic Republic' 'Mexico' 'New Zealand' 'Oman' 'Peru'  
'Republic of Korea' 'Republic of North Macedonia' 'Russian Federation'  
'Saint Lucia' 'Samoa' 'Saudi Arabia' 'Senegal' 'Sierra Leone' 'Singapore'  
'Slovakia' 'Sweden' 'Thailand' 'Tonga' 'Turkmenistan'  
'United Arab Emirates' 'United States of America'  
'Venezuela (Bolivarian Republic of)' 'Yemen']
```

```
Forma de x_train: (1819, 23)
```

```
Forma de y_train: (1819,)
```

```
Forma de x_val: (595, 23)
```

```
Forma de y_val: (595,)
```

```
Forma de x_test: (629, 23)
```

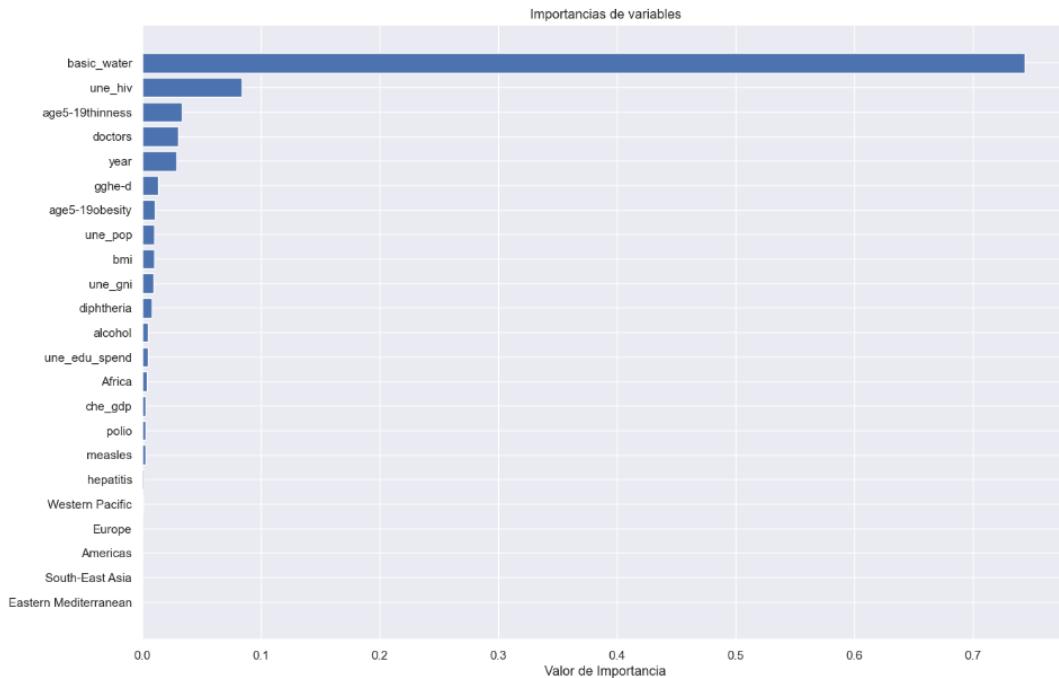
```
Forma de y_test: (629,)
```

3.3.4.2 Entrenamiento del modelo

```
gb_fs = GradientBoostingRegressor(random_state = semilla, max_depth = 10).fit(x_train, y_train)
```

3.3.4.3 Importancia de las características

Los métodos basados en arboles de decisión funcionan buscando las variables que más reducen la entropía, o, en otras palabras, la que mayor ganancia de información representa. El atributo "feature_importances_" calculado por estos modelos permite identificar las variables que suelen ser más propensas a usarse por tener más ganancia. Esto lo logra promediando la "importancia" a través de todos los splits de todos los árboles.

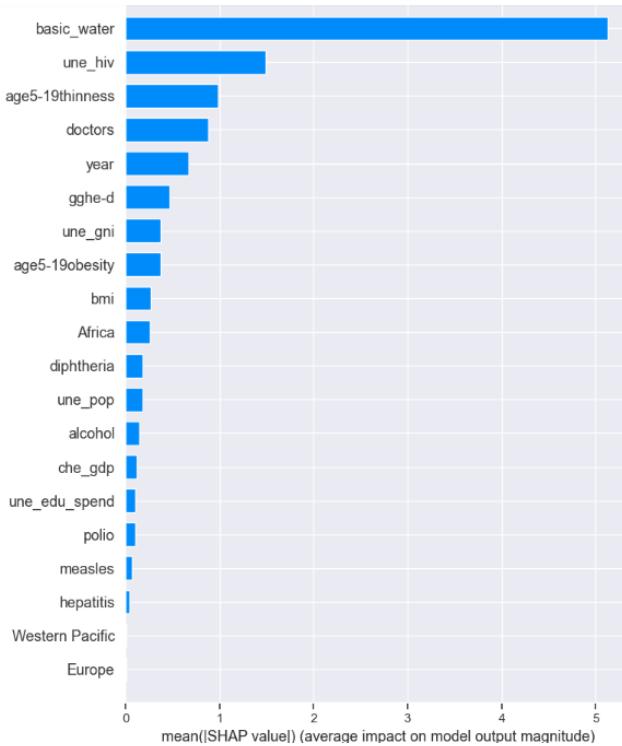


Del gráfico se puede extraer que la variable con mayor importancia para el modelo fue basic_water. La diferencia en importancias es realmente significativa al comparar esta variable con las demás. Por ende, resulta evidente que basic_water es la variable que más información aporta al modelo.

Por otra parte, se comprueba, en concordancia con lo encontrado en la matriz de correlación parcial, que todas las variables de región, a excepción de Africa, no son importantes para el modelo.

3.3.5 Valores SHAP

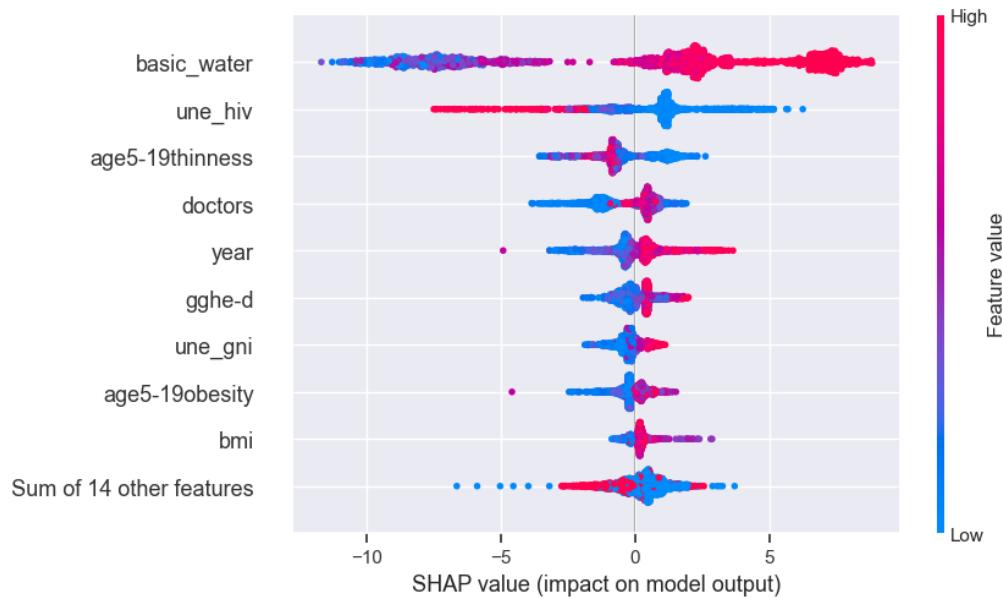
Los valores de Shapley (valores SHAP), originarios de la teoría de juegos (juegos cooperativos), nos permiten, en el campo de la ciencia de datos y machine learning, entender cómo las características individuales contribuyen a las predicciones de los modelos. En otras palabras, su importancia para el modelo.



Estos resultados son en gran medida similares a los obtenidos con el método previo. En este caso, la variable basic_water sigue siendo la de mayor importancia. Sin embargo, a diferencia del método anterior, la importancia de basic_water no es tan desproporcionada en comparación con las demás variables.

La interpretación que se le puede dar al valor de SHAP de aproximadamente 5.1 de basic_water es que, en promedio, la disponibilidad de agua potable tiene un impacto de 5.1 años en la predicción de la esperanza de vida al nacer en los países analizados. Esta misma explicación engloba las demás variables.

Asimismo, en este método también se evidencia la carencia de importancia que tienen las variables de región.



Este gráfico señala la dirección en la que cada variable impacta el modelo. En el caso de la variable basic_water, los valores mayores de porcentaje de la población con acceso a servicios básicos de agua potable están asociados a una mayor esperanza de vida, por su aporte positivo a la predicción. Por el contrario, los valores bajos e intermedios aportan negativamente a la esperanza de vida.

El impacto de las demás variables mostradas es en gran medida acordes a las apreciaciones realizadas en las matrices de correlación, en su mayoría presentando una división marcada en el tipo de impacto que tienen. Sin embargo, en variables como age5-19thinness, doctors, y gghe-d, se encuentran algunos valores atípicos.

Debido a la agrupación de las demás variables en una sola categoría, no se pueden hacer juicios precisos sobre estas.

3.3.6 Selección de variables con Backward Selection

Para la selección de variables con el método de backward selection, inicialmente se evalúa el desempeño del modelo inicial, entrenado con todas las variables disponibles.

En cada iteración se elimina una variable que, con base a los análisis realizados previamente, no es de importancia para el modelo y puede ser omitida.

```
pred_val_gb_fs = gb_fs.predict(x_val)

MAE0 = mean_absolute_error(y_val, pred_val_gb_fs)
print("MAE: " + str(MAE0))
```

MAE: 2.8396271408898834

3.3.6.1 Iteración 1

```
x_train1 = x_train.drop(columns=["Eastern Mediterranean"])
x_val1 = x_val.drop(columns=["Eastern Mediterranean"])

gb_fs1 = GradientBoostingRegressor(random_state = semilla, max_depth = 10).fit(x_train1, y_train)

pred_val_gb_fs1 = gb_fs1.predict(x_val1)

MAE1 = mean_absolute_error(y_val, pred_val_gb_fs1)
print("MAE: " + str(MAE1))
```

MAE: 2.868043536919937

```
MAE1-MAE0
```

0.028416396030053637

3.3.6.2 Iteración 2

```
x_train2 = x_train.drop(columns=["Eastern Mediterranean", "South-East Asia"])
x_val2 = x_val.drop(columns=["Eastern Mediterranean", "South-East Asia"])

gb_fs2 = GradientBoostingRegressor(random_state = semilla, max_depth = 10).fit(x_train2, y_train)

pred_val_gb_fs2 = gb_fs2.predict(x_val2)

MAE2 = mean_absolute_error(y_val, pred_val_gb_fs2)
print("MAE: "+ str(MAE2))

MAE: 2.837373083718009
```

MAE2-MAE1

-0.030670453201928183

3.3.6.3 Iteración 3

```
x_train3 = x_train.drop(columns=["Eastern Mediterranean", "South-East Asia", "Americas"])
x_val3 = x_val.drop(columns=["Eastern Mediterranean", "South-East Asia", "Americas"])

gb_fs3 = GradientBoostingRegressor(random_state = semilla, max_depth = 10).fit(x_train3, y_train)

pred_val_gb_fs3 = gb_fs3.predict(x_val3)

MAE3 = mean_absolute_error(y_val, pred_val_gb_fs3)
print("MAE: "+ str(MAE3))

MAE: 2.8215049717803273
```

MAE3-MAE2

-0.01586811193768156

3.3.6.4 Iteración 4

```
x_train4 = x_train.drop(columns=["Eastern Mediterranean", "South-East Asia", "Americas", "Europe"])
x_val4 = x_val.drop(columns=["Eastern Mediterranean", "South-East Asia", "Americas", "Europe"])

gb_fs4 = GradientBoostingRegressor(random_state = semilla, max_depth = 10).fit(x_train4, y_train)

pred_val_gb_fs4 = gb_fs4.predict(x_val4)

MAE4 = mean_absolute_error(y_val, pred_val_gb_fs4)
print("MAE: "+ str(MAE4))

MAE: 2.8341545590825223
```

MAE4-MAE3

0.012649587302195009

3.3.6.5 Iteración 5

```
x_train5 = x_train.drop(columns=["Eastern Mediterranean", "South-East Asia", "Americas", "Europe", "Western Pacific"])
x_val5 = x_val.drop(columns=["Eastern Mediterranean", "South-East Asia", "Americas", "Europe", "Western Pacific"])

gb_fs5 = GradientBoostingRegressor(random_state = semilla, max_depth = 10).fit(x_train5, y_train)

pred_val_gb_fs5 = gb_fs5.predict(x_val5)

MAE5 = mean_absolute_error(y_val, pred_val_gb_fs5)
print("MAE: " + str(MAE5))

MAE: 2.8402136283911172
```

MAE5-MAE4

0.0060590693085949

MAE5-MAE0

0.0005864875012338011

3.3.6.6 Iteración 6

```
x_train6 = x_train.drop(columns=["Eastern Mediterranean", "South-East Asia", "Americas", "Europe", "Western Pacific",
                                 "hepatitis"])
x_val6 = x_val.drop(columns=["Eastern Mediterranean", "South-East Asia", "Americas", "Europe", "Western Pacific", "hepatitis"])

gb_fs6 = GradientBoostingRegressor(random_state = semilla, max_depth = 10).fit(x_train6, y_train)

pred_val_gb_fs6 = gb_fs6.predict(x_val6)

MAE6 = mean_absolute_error(y_val, pred_val_gb_fs6)
print("MAE: " + str(MAE6))
```

MAE: 2.918020206861821

MAE6-MAE5

0.0778065784707036

MAE6-MAE0

0.0783930659719374

3.3.6.7 Iteración 7

```
x_train7 = x_train.drop(columns=["Eastern Mediterranean", "South-East Asia", "Americas", "Europe", "Western Pacific",
                                 "hepatitis", "measles"])
x_val7 = x_val.drop(columns=["Eastern Mediterranean", "South-East Asia", "Americas", "Europe", "Western Pacific", "hepatitis",
                            "measles"])

gb_fs7 = GradientBoostingRegressor(random_state = semilla, max_depth = 10).fit(x_train7, y_train)

pred_val_gb_fs7 = gb_fs7.predict(x_val7)

MAE7 = mean_absolute_error(y_val, pred_val_gb_fs7)
print("MAE: " + str(MAE7))
```

MAE: 2.966875323966966

MAE7-MAE6

0.04885511710514523

MAE7-MAE0

0.12724818307708263

Los resultados indican que, a partir de la 5ta iteración, el desempeño del modelo comienza a decaer significativamente. En la eliminación paulatina de las variables regionales, el máximo aumento en MAE, de 0.028416396030053637, ocurrió en la primera iteración. Sin embargo, la eliminación de todas las variables de

región, menos Africa, solo representaron un aumento de 0.0005864875012338011 en el MAE con respecto al MAE inicial del modelo.

En la 6ta iteración, la eliminación de la variable hepatitis produce un aumento de 0.0783930659719374 en el MAE con respecto al inicial. Por lo tanto, con toda la evidencia recopilada en el backward selection y en los métodos anteriores, resulta clara la falta de importancia de las variables Eastern Mediterranean, South-East Asia, Americas, Europe, y Western Pacific, para las predicciones del modelo.

3.4 División de datos seleccionados en entrenamiento, validación y prueba

Con base en el proceso de backward selection efectuado, se crean nuevos conjuntos de entrenamiento, validación y prueba, omitiendo las variables que no aportan información nueva para el modelo o tienen efectos negativos o insignificantes en este.

3.5 PCA

El Análisis de Componentes Principales (PCA, por sus siglas en inglés) es una técnica de reducción de dimensionalidad que se utiliza para identificar patrones en datos de alta dimensionalidad y expresar esos datos en forma de componentes principales no correlacionados.

El objetivo principal del PCA es transformar un conjunto de variables correlacionadas en un nuevo conjunto de variables no correlacionadas llamadas componentes principales. Estos componentes principales están ordenados de tal manera que el primero captura la mayor variabilidad presente en los datos, el segundo captura la segunda mayor variabilidad, y así sucesivamente.

El proceso general del PCA implica los siguientes pasos:

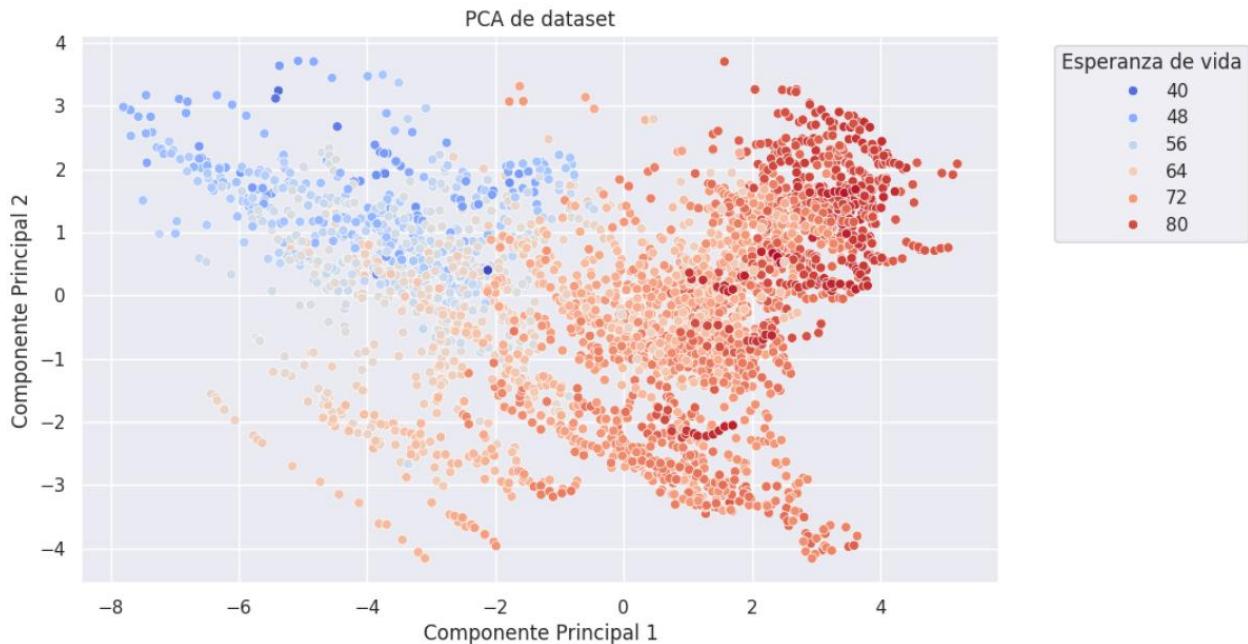
1. **Estandarizar los datos:** Se transforman los datos para que estos tengan una media de 0 y una desviación estándar de 1.
2. **Calcular la matriz de covarianza:** La matriz de covarianza mide cómo cambian dos variables juntas. La matriz de covarianza es una matriz cuadrada donde cada elemento (i,j) representa la covarianza entre la i-ésima y la j-ésima variable.
3. **Cálculo de los Valores y Vectores Propios con SVD:** Se calculan los valores propios (eigenvalues) y los vectores propios (eigenvectors) de la matriz de covarianza. Los valores propios indican la cantidad de varianza explicada por cada componente principal, y los vectores propios definen la dirección de cada componente principal.
4. **Selección de componentes principales:** Se ordenan los vectores propios de acuerdo a sus valores propios asociados. Los primeros vectores propios, que corresponden a los valores propios más grandes, son los componentes principales más importantes.
5. **Proyección de datos:** Se proyectan los datos originales en el espacio definido por los componentes principales seleccionados. Esto transforma los datos originales en un nuevo conjunto de datos de menor dimensionalidad.

3.5.1 PCA para 2 componentes principales

Para realizar el Análisis de Componentes Principales sobre este dataset, inicialmente se estandarizan los datos numéricos. Sobre este nuevo conjunto de datos, se aplica la herramienta de PCA de sklearn, obteniendo los siguientes resultados para los primeros 2 componentes principales:

	Varianza explicada por cada componente: [0.32411521 0.10400957]	
	PC1	PC2
year	0.069390	-0.100328
une_pop	-0.037910	-0.129605
une_gni	0.214721	-0.067934
gghe-d	0.266757	0.276492
che_gdp	0.162204	0.346987
une_edu_spend	0.125815	0.138423
basic_water	0.305107	-0.126266
doctors	0.284867	0.167286
bmi	0.280149	-0.079511
age5-19thinness	-0.276021	-0.182779
age5-19obesity	0.249684	-0.196769
alcohol	0.196193	0.353466
hepatitis	0.191704	-0.215288
measles	0.284226	-0.186761
polio	0.283508	-0.173851
diphtheria	0.282407	-0.176423
une_hiv	-0.108972	0.171325
Africa	-0.246707	0.204012
Americas	0.081604	-0.076831
Eastern Mediterranean	0.003968	-0.334733
Europe	0.201154	0.314070
South-East Asia	-0.083179	-0.241318
Western Pacific	0.011773	-0.129709

Gráficamente, estos presentan la siguiente distribución:



Con base en estos resultados podemos sacar las siguientes conclusiones:

1. **Varianza explicada por cada componente:**

- **Componente Principal 1 (PC1):** Explica aproximadamente el 32.41% de la varianza total en los datos.
- **Componente Principal 2 (PC2):** Explica aproximadamente el 10.40% de la varianza total en los datos.
- **Interpretación:** Las dos primeras componentes principales juntas explican aproximadamente el 42.82% de la varianza total. Esto sugiere que casi la mitad de la información presente en los datos originales puede ser representada en un espacio de 2 dimensiones.

2. **Cargas de las componentes principales:** Las cargas de las componentes principales indican la contribución de cada variable original a las componentes principales.

- **PC1:**
 - **Basic_water (-0.305107):** Esta variable tiene una alta contribución negativa a PC1. Es decir, los países con acceso a agua básica están más alejados en la dirección negativa de PC1.
 - **Doctors (-0.284867), Measles (-0.284226), Polio (-0.283508), Diphtheria (-0.282407):** Estas variables también tienen una alta contribución negativa, indicando que países con más doctores y mejores coberturas de vacunación están en la dirección negativa de PC1.
 - **age5-19thinness (0.276021) y Africa (0.246707):** Las variables 'age5-19thinness' y 'Africa' tienen una alta contribución positiva, sugiriendo que los países con mayor porcentaje de delgadez y que pertenecen a la región de África se encuentran más hacia la dirección positiva de PC1.
- **PC2:**
 - **Eastern Mediterranean (0.334733):** Esta variable tiene una alta contribución positiva a PC2. Esto sugiere que los países de esta región están más alejados en la dirección positiva de PC2.
 - **Che_gdp (-0.346987) y Alcohol (-0.353466):** Estas variables tienen una alta contribución negativa a PC2, sugiriendo que países con un mayor gasto en salud en relación al PIB y mayor consumo de alcohol están en la dirección negativa de PC2.

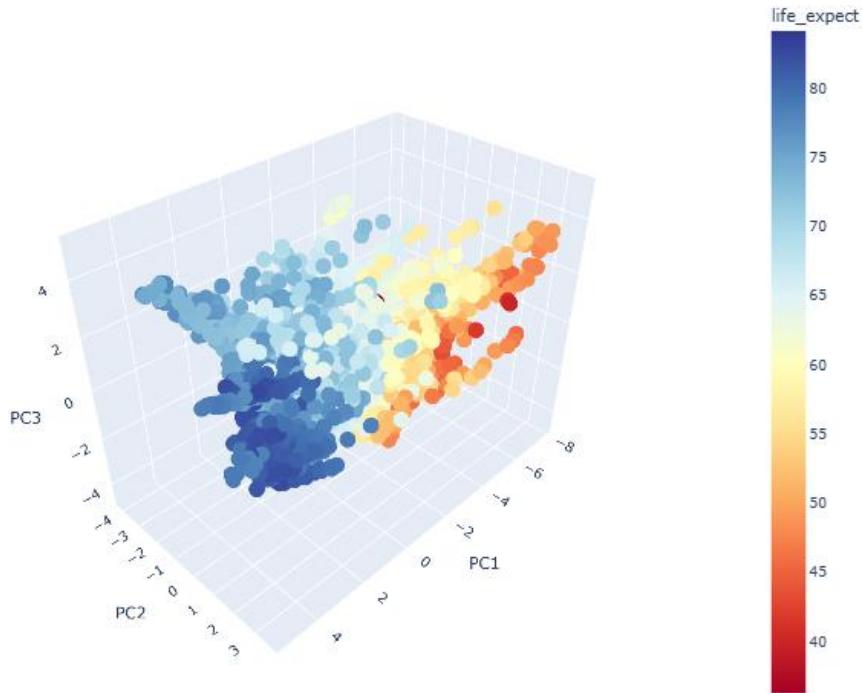
3. **Visualización:** El gráfico muestra cómo los países se agrupan en el espacio definido por las dos primeras componentes principales. El color de los puntos representa la esperanza de vida, con tonos más cálidos (rojo) indicando una mayor esperanza de vida y tonos más fríos (azul) indicando una menor esperanza de vida.

- **Agrupamientos:** Podemos observar que hay una tendencia en la cual los países con mayor esperanza de vida están agrupados en la región negativa de PC1 y a lo largo de todo PC2, mientras que los países con menor esperanza de vida están más hacia la región positiva de PC1 y en la región negativa de PC2.
- **Dispersión:** La dispersión de los puntos indica la variabilidad en los datos. Los puntos más dispersos sugieren una mayor variabilidad en los valores de las variables originales.

3.5.2 PCA para 3 componentes principales

Para los primeros 3 componentes principales se obtuvieron los siguientes resultados:

	Varianza explicada por cada componente: [0.32411521 0.18400957 0.08186014]		
	PC1	PC2	PC3
year	0.069390	-0.100328	-0.001682
une_pop	-0.037910	-0.129605	0.148016
une_gni	0.214721	-0.067934	0.166889
ghe_d	0.266757	0.276492	0.029285
che_gdp	0.162204	0.346987	0.001162
une_edu_spend	0.125815	0.138423	0.043080
basic_water	0.305107	-0.126265	-0.014481
doctors	0.284867	0.167286	-0.005045
bmi	0.280149	-0.079511	0.304411
ages-19thinness	-0.276021	-0.182779	-0.207350
ages-19obesity	0.249684	-0.196769	0.362985
alcohol	0.196193	0.353466	-0.115242
hepatitis	0.191704	-0.215288	-0.212886
measles	0.284226	-0.186761	-0.303726
polio	0.283508	-0.173851	-0.333371
diphtheria	0.282407	-0.176423	-0.339585
une_hiv	-0.108972	0.171325	-0.149661
Africa	-0.246707	0.204012	-0.113093
Americas	0.081604	-0.076831	0.169562
Eastern Mediterranean	0.003968	0.334733	0.290520
Europe	0.201154	0.314070	-0.157850
South-East Asia	-0.083179	-0.241318	-0.348374
Western Pacific	0.011773	-0.129709	0.134160



Para 3 o más componentes principales, el análisis es igual al realizado para 2 componentes.

En este caso, la tercera componente principal explica aproximadamente el 8.19% de la varianza total en los datos. Por lo tanto, las 3 componentes en conjunto explican el 50.99% de los datos.

Igualmente, para la PC3, la variable "age5-19obesity" tiene una alta contribución positiva con 0.362940, y Polio y Diphtheria tienen altas contribuciones negativas con -0.333371 y -0.335989 respectivamente.

Por último, se puede observar que a medida que aumentan los componentes principales, más difícil es identificar como estos se distribuyen con respecto a la variable objetivo y como cada componente principal se relaciona con las demás.

3.5.3 PCA para 6 componentes principales

Finalmente, se obtuvieron los siguientes resultados para los primeros 6 componentes principales:

```
Varianza explicada por cada componente: [0.32411521 0.10400957 0.08186014 0.07796809 0.05958547 0.05741641]
Varianza total explicada por las 6 componentes: 70.49548976838746%
```

Con los componentes principales que explican más del 70% de la varianza de los datos, se evaluará el desempeño de un modelo entrenado con estas componentes, y se comparará con los demás modelos entrenados con todas las variables y con las variables seleccionadas en el backward selection.

3.5.4 División de componentes principales en entrenamiento, validación y prueba

Se crean nuevos conjuntos de entrenamiento, validación y prueba compuestos por los primeros 6 componentes principales.

```
Forma de x_train_pca: (1819, 6)
Forma de x_val_pca: (595, 6)
Forma de x_test_pca: (629, 6)
```

4. Modelación

En esta etapa se entrenarán 27 modelos, 3 por cada uno de los siguientes modelos: regresión lineal múltiple, regresión de Lasso, regresión de Ridge, regresión de ElasticNet, regresión de vectores de soporte, regresión de k-vecinos más cercanos, árbol de decisión, random forest, y gradient boosting. De cada conjunto de 3, 1 será entrenado y evaluado con los conjuntos de entrenamiento y validación que cuentan con todas las variables, otro con los conjuntos que contienen solo las variables seleccionadas por backward selection, y el último con los conjuntos que contienen los primeros 6 componentes principales. La evaluación se hará sobre las métricas de MSE, MAE, MAPE, y Score (R^2).

A fin de seleccionar los modelos con mejor rendimiento, se llevará a cabo una optimización de hiperparámetros con la herramienta RandomizedSearchCV. Como lo indica Elyse Lee en el blog Medium (2019), el método Random Search utiliza combinaciones aleatorias de hiperparámetros. Por lo tanto, a diferencia del método Grid Search, no se prueban todos los valores de los parámetros y, en su lugar, se muestran con un número fijo de iteraciones. La utilización de un método sobre el otro busca reducir el costo computacional y la duración del proceso.

Para cada uno de los modelos, a excepción de la regresión lineal múltiple, para la cual se usa un `n_iter = 4`, se especifica un total de 100 iteraciones en el parámetro "`n_iter`", y 3 pliegues para el parámetro de validación cruzada "`cv`". Esto implica que para cada una de las 100 combinaciones de hiperparámetros, se ejecutará una validación cruzada de 3 pliegues. En otras palabras, el modelo se entrenará y evaluará 3 veces por cada combinación, resultando en un total de 300 entrenamientos y evaluaciones (100 combinaciones \times 3 pliegues por combinación).

En adición a los modelos de regresión, se entrenarán 3 modelos de regresión logística multinomial para clasificación supervisada con los 3 conjuntos de entrenamiento utilizados para los modelos de regresión. La evaluación se hará sobre las métricas de precision, recall, f1-score y accuracy. De igual forma, se empleará la herramienta RandomizedSearchCV con los mismos hiperparámetros empleados en los modelos de regresión.

Por último, se entrenará un modelo de KMeans para la clasificación no supervisada de los países según sus características del 2016 y comparando los clusters generados con la esperanza de vida real para este mismo año.

4.1 Multiple Linear Regression

Se fija la siguiente malla de hiperparámetros:

```
lr = LinearRegression(random_state=semilla)

param_grid = {
    'fit_intercept': [True, False],
    'positive': [True, False]
}
```

Para cada uno de los modelos se ejecuta el mismo tipo de entrenamiento con RandomizedSearchCV, pero con sus respectivos conjuntos de entrenamiento y validación:

```
Fitting 3 folds for each of 4 candidates, totalling 12 fits
RandomizedSearchCV
RandomizedSearchCV(cv=3, estimator=LinearRegression(), n_iter=4, n_jobs=-1,
   param_distributions={'fit_intercept': [True, False],
                        'positive': [True, False]},
   random_state=42, verbose=2)
   best_estimator_:
       LinearRegression
           LinearRegression
```

4.1.1 Entrenamiento con todas las variables

Obteniendo los siguientes mejores parámetros:

```
{'positive': False, 'fit_intercept': True}
```

Y los siguientes resultados:

```
MSE: 9.976303182341352
MAE: 2.533060773095239
MAPE: 0.0382966065975254
R2: 0.8973358999925602
```

4.1.2 Entrenamiento con variables seleccionadas

Obteniendo los siguientes mejores parámetros:

```
{'positive': True, 'fit_intercept': True}
```

Y los siguientes resultados:

```
MSE: 20.175969118566126
MAE: 3.212915176473855
MAPE: 0.050544691065227944
R2: 0.7913163069579529
```

4.1.3 Entrenamiento con componentes principales

Obteniendo los siguientes mejores parámetros:

```
{'positive': True, 'fit_intercept': True}
```

Y los siguientes resultados:

```
MSE: 26.726849058270496
MAE: 3.964034006363923
MAPE: 0.06157189812704891
R2: 0.7812099121050492
```

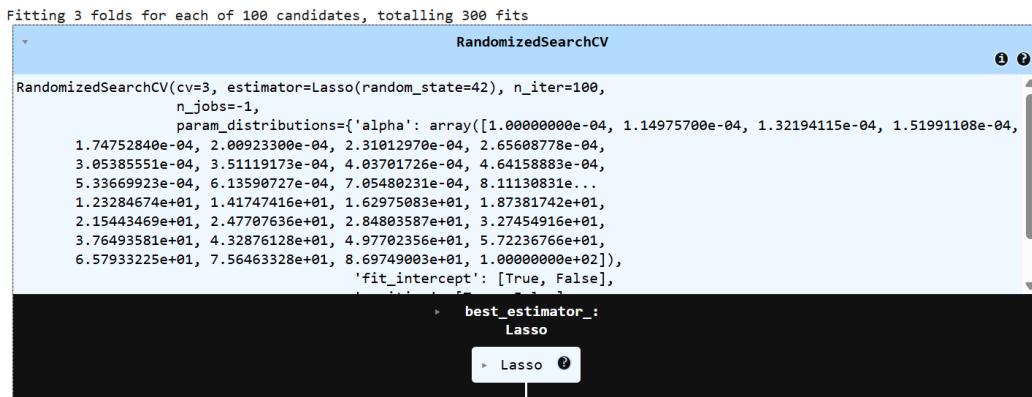
4.2 Lasso Regression

Se fija la siguiente malla de hiperparámetros:

```
lasso = Lasso(random_state=semilla)

param_grid = {
    'alpha': np.logspace(-4, 2, 100),
    'fit_intercept': [True, False],
    'selection': ['cyclic', 'random'],
    'positive': [True, False]
}
```

Para cada uno de los modelos se ejecuta el mismo tipo de entrenamiento con RandomizedSearchCV, pero con sus respectivos conjuntos de entrenamiento y validación:



4.2.1 Entrenamiento con todas las variables

Obteniendo los siguientes mejores parámetros:

```
{'selection': 'cyclic',
'positive': False,
'fit_intercept': True,
'alpha': np.float64(0.040370172585965536)}
```

Y los siguientes resultados:

MSE: 8.966080621230422
MAE: 2.4437954104713433
MAPE: 0.03684337742526799
R²: 0.8954585748581578

4.2.2 Entrenamiento con variables seleccionadas

Obteniendo los siguientes mejores parámetros:

```
{'selection': 'cyclic',
 'positive': False,
 'fit_intercept': True,
 'alpha': np.float64(0.040370172585965536)}
```

Y los siguientes resultados:

MSE: 8.854379969551498
MAE: 2.416209461852399
MAPE: 0.03641714900213478
R²: 0.8945946760405484

4.2.3 Entrenamiento con componentes principales

Obteniendo los siguientes mejores parámetros:

```
{'selection': 'random',
 'positive': False,
 'fit_intercept': True,
 'alpha': np.float64(0.093260334688322)}
```

Y los siguientes resultados:

MSE: 11.128057115287707
MAE: 2.7400979292522556
MAPE: 0.04134219672347985
R²: 0.8579530736431715

4.3 Ridge Regression

Se fija la siguiente malla de hiperparámetros:

```
ridge = Ridge(random_state=semilla)

param_grid = {
    'alpha': np.logspace(-4, 4, 100),
    'fit_intercept': [True, False],
    'solver': ['auto', 'svd', 'cholesky', 'lsqr', 'sparse_cg', 'sag', 'saga'],
    'positive': [True, False]
}
```

Para cada uno de los modelos se ejecuta el mismo tipo de entrenamiento con RandomizedSearchCV, pero con sus respectivos conjuntos de entrenamiento y validación:

```
Fitting 3 folds for each of 100 candidates, totalling 300 fits
    ▾ RandomizedSearchCV
      RandomizedSearchCV(cv=3, estimator=Ridge(random_state=42), n_iter=100,
          n_jobs=-1,
          param_distributions={'alpha': array([1.0000000e-04, 1.20450354e-04, 1.45082878e-04, 1.74752840e-04,
2.10490414e-04, 2.53536449e-04, 3.05385551e-04, 3.67837977e-04,
4.43062146e-04, 5.33669923e-04, 6.42807312e-04, 7.74263683e-04,
9.32603347e-04, 1.12332403e-03, 1.35304777e-03, 1.62975083e...
6.13590727e+02, 7.39072203e+02, 8.90215085e+02, 1.07226722e+03,
1.29154967e+03, 1.55567614e+03, 1.87381742e+03, 2.25701972e+03,
2.71858824e+03, 3.27454916e+03, 3.94420606e+03, 4.75081016e+03,
5.72236766e+03, 6.89261210e+03, 8.30217568e+03, 1.0000000e+04]),
          'fit_intercept': [True, False],
          'n_iter': [5, 10, 20, 50, 100]
      } best_estimator_:
        Ridge
      ↴ Ridge
```

4.3.1 Entrenamiento con todas las variables

Obteniendo los siguientes mejores parámetros:

```
{'solver': 'sag',
 'positive': False,
 'fit_intercept': True,
 'alpha': np.float64(31.257158496882415)}
```

Y los siguientes resultados:

```
MSE: 9.63333657401883
MAE: 2.515567701051771
MAPE: 0.03804235902487542
R2: 0.8968517606367876
```

4.3.2 Entrenamiento con variables seleccionadas

Obteniendo los siguientes mejores parámetros:

```
{'solver': 'sag',
 'positive': False,
 'fit_intercept': True,
 'alpha': np.float64(31.257158496882415)}
```

Y los siguientes resultados:

```
MSE: 8.987836789868954
MAE: 2.431888824372141
MAPE: 0.03675879786778279
R2: 0.8948839392541248
```

4.3.3 Entrenamiento con componentes principales

Obteniendo los siguientes mejores parámetros:

```

{'solver': 'lsqr',
 'positive': False,
 'fit_intercept': True,
 'alpha': np.float64(242.01282647943833)}

```

Y los siguientes resultados:

MSE: 11.29618702911104
 MAE: 2.758468573542848
 MAPE: 0.04171429574477086
 R²: 0.8575821627229226

4.4 ElasticNet Regression

Se fija la siguiente malla de hiperparámetros:

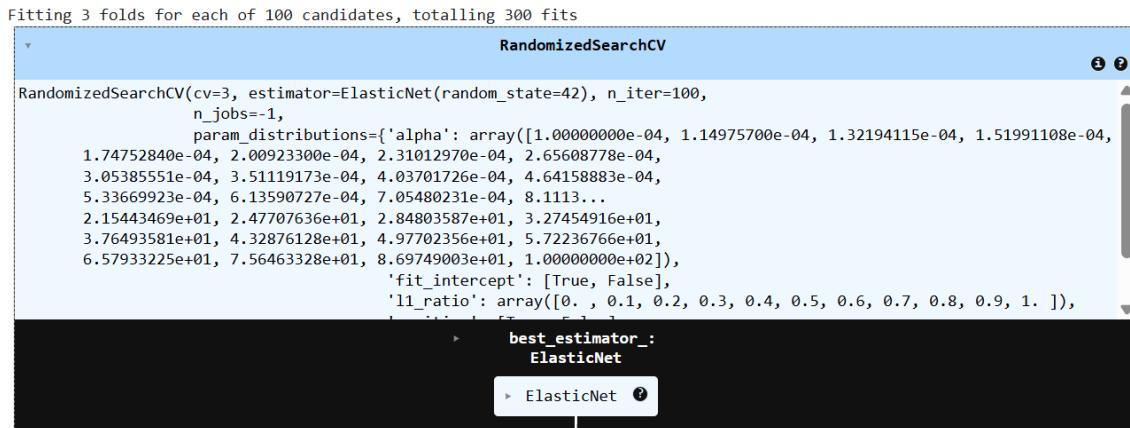
```

enet = ElasticNet(random_state=semilla)

param_grid = {
    'alpha': np.logspace(-4, 2, 100),
    'l1_ratio': np.linspace(0, 1, 11),
    'fit_intercept': [True, False],
    'selection': ['cyclic', 'random'],
    'positive': [True, False]
}

```

Para cada uno de los modelos se ejecuta el mismo tipo de entrenamiento con RandomizedSearchCV, pero con sus respectivos conjuntos de entrenamiento y validación:



```

Fitting 3 folds for each of 100 candidates, totalling 300 fits
RandomizedSearchCV
RandomizedSearchCV(cv=3, estimator=ElasticNet(random_state=42), n_iter=100,
n_jobs=-1,
param_distributions={'alpha': array([1.00000000e-04, 1.14975700e-04, 1.32194115e-04, 1.51991108e-04,
1.74752840e-04, 2.00923300e-04, 2.31012970e-04, 2.65608778e-04,
3.05385551e-04, 3.51119173e-04, 4.03701726e-04, 4.64158883e-04,
5.33669923e-04, 6.13590727e-04, 7.05480231e-04, 8.1113...
2.15443469e+01, 2.47707636e+01, 2.84803587e+01, 3.27454916e+01,
3.76493581e+01, 4.32876128e+01, 4.97702356e+01, 5.72236766e+01,
6.57933225e+01, 7.56463328e+01, 8.69749003e+01, 1.00000000e+02]),
'fit_intercept': [True, False],
'l1_ratio': array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
'selection': ['cyclic', 'random'],
'positive': [True, False]}
    best_estimator_:
        ElasticNet
            ElasticNet

```

4.4.1 Entrenamiento con todas las variables

Obteniendo los siguientes mejores parámetros:

```

{'selection': 'random',
 'positive': False,
 'l1_ratio': np.float64(0.3000000000000004),
 'fit_intercept': True,
 'alpha': np.float64(0.040370172585965536)}

```

Y los siguientes resultados:

```
MSE: 9.654355281209039  
MAE: 2.5157236495246504  
MAPE: 0.03803979756185196  
R2: 0.8969666989549435
```

4.4.2 Entrenamiento con variables seleccionadas

Obteniendo los siguientes mejores parámetros:

```
{'selection': 'random',  
 'positive': False,  
 'fit_intercept': True,  
 'alpha': np.float64(0.026560877829466867)}
```

Y los siguientes resultados:

```
MSE: 8.940844974436024  
MAE: 2.4277188590746532  
MAPE: 0.03667070716146991  
R2: 0.8948103886942881
```

4.4.3 Entrenamiento con componentes principales

Obteniendo los siguientes mejores parámetros:

```
{'selection': 'random',  
 'positive': False,  
 'fit_intercept': True,  
 'alpha': np.float64(0.093260334688322)}
```

Y los siguientes resultados:

```
MSE: 11.204969893848888  
MAE: 2.7498050255244864  
MAPE: 0.04150848058639663  
R2: 0.8579685554863127
```

4.5 Support Vector Regression

Se fija la siguiente malla de hiperparámetros:

```

svr = SVR()

param_grid = {
    'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
    'C': [0.1, 1, 10, 100, 1000],
    'epsilon': [0.001, 0.01, 0.1, 0.2, 0.5],
    'degree': [2, 3, 4, 5],
    'gamma': ['scale', 'auto']
}

```

Para cada uno de los modelos se ejecuta el mismo tipo de entrenamiento con RandomizedSearchCV, pero con sus respectivos conjuntos de entrenamiento y validación:

```

Fitting 3 folds for each of 100 candidates, totalling 300 fits
RandomizedSearchCV
RandomizedSearchCV(cv=3, estimator=SVR(), n_iter=100, n_jobs=-1,
                    param_distributions={'C': [0.1, 1, 10, 100, 1000],
                                         'degree': [2, 3, 4, 5],
                                         'epsilon': [0.001, 0.01, 0.1, 0.2, 0.5],
                                         'gamma': ['scale', 'auto'],
                                         'kernel': ['linear', 'poly', 'rbf',
                                                    'sigmoid']},
                    random_state=42, verbose=2)
    > best_estimator_:
        SVR
            > SVR

```

4.5.1 Entrenamiento con todas las variables

Obteniendo los siguientes mejores parámetros:

```
{'kernel': 'linear', 'gamma': 'scale', 'epsilon': 0.5, 'degree': 3, 'C': 0.1}
```

Y los siguientes resultados:

```

MSE: 10.681391177316396
MAE: 2.598306817944479
MAPE: 0.03946948127489532
Score: 0.8936733433480438

```

4.5.2 Entrenamiento con variables seleccionadas

Obteniendo los siguientes mejores parámetros:

```
{'kernel': 'linear', 'gamma': 'scale', 'epsilon': 0.5, 'degree': 3, 'C': 0.1}
```

Y los siguientes resultados:

```

MSE: 9.93175880393883
MAE: 2.523683302235687
MAPE: 0.0383781470705669
Score: 0.8920009729995876

```

4.5.3 Entrenamiento con componentes principales

Obteniendo los siguientes mejores parámetros:

```
{'kernel': 'linear', 'gamma': 'scale', 'epsilon': 0.1, 'degree': 4, 'C': 0.1}
```

Y los siguientes resultados:

```
MSE: 12.681480174202479  
MAE: 2.8932207225534348  
MAPE: 0.04373286256900464  
Score: 0.8542697899327842
```

4.6 K-Nearest Neighbors Regression

Se fija la siguiente malla de hiperparámetros:

```
knn = KNeighborsRegressor()  
  
param_grid = {  
    'n_neighbors': [3, 5, 7, 9, 11, 13, 15],  
    'weights': ['uniform', 'distance'],  
    'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],  
    'leaf_size': [10, 20, 30, 40],  
    'p': [1, 2],  
    'metric': ['minkowski', 'euclidean', 'manhattan']  
}
```

Para cada uno de los modelos se ejecuta el mismo tipo de entrenamiento con RandomizedSearchCV, pero con sus respectivos conjuntos de entrenamiento y validación:

```
Fitting 3 folds for each of 100 candidates, totalling 300 fits  
RandomizedSearchCV  
RandomizedSearchCV(cv=3, estimator=KNeighborsRegressor(), n_iter=100, n_jobs=-1,  
    param_distributions={'algorithm': ['auto', 'ball_tree',  
        'kd_tree', 'brute'],  
        'leaf_size': [10, 20, 30, 40],  
        'metric': ['minkowski', 'euclidean',  
            'manhattan'],  
        'n_neighbors': [3, 5, 7, 9, 11, 13, 15],  
        'p': [1, 2],  
        'weights': ['uniform', 'distance']},  
    random_state=42, verbose=2)  
    best_estimator_:  
    KNeighborsRegressor  
    KNeighborsRegressor ?
```

4.6.1 Entrenamiento con todas las variables

Obteniendo los siguientes mejores parámetros:

```
{'weights': 'distance',
 'p': 2,
 'n_neighbors': 15,
 'metric': 'manhattan',
 'leaf_size': 30,
 'algorithm': 'kd_tree'}
```

Y los siguientes resultados:

```
MSE: 9.802632044540829
MAE: 2.4879109809599695
MAPE: 0.03853221178889343
Score: 1.0
```

4.6.2 Entrenamiento con variables seleccionadas

Obteniendo los siguientes mejores parámetros:

```
{'weights': 'distance',
 'p': 2,
 'n_neighbors': 15,
 'metric': 'manhattan',
 'leaf_size': 30,
 'algorithm': 'kd_tree'}
```

Y los siguientes resultados:

```
MSE: 9.846772530475988
MAE: 2.5271456835439317
MAPE: 0.039143887434272585
Score: 1.0
```

4.6.3 Entrenamiento con componentes principales

Obteniendo los siguientes mejores parámetros:

```
{'weights': 'distance',
 'p': 1,
 'n_neighbors': 15,
 'metric': 'euclidean',
 'leaf_size': 30,
 'algorithm': 'auto'}
```

Y los siguientes resultados:

```
MSE: 11.633290226800314
MAE: 2.697043061988626
MAPE: 0.04117031683103427
Score: 1.0
```

4.7 Decision Tree

Se fija la siguiente malla de hiperparámetros:

```
dt = DecisionTreeRegressor(random_state = semilla)

criterion = ['squared_error', 'absolute_error', 'friedman_mse', 'poisson']
splitter = ['best', 'random']
max_features = ['auto', 'sqrt','log2']
max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
max_depth.append(None)
min_samples_split = [2, 5, 10]
min_samples_leaf = [1, 2, 4]

param_grid = {'criterion': criterion,
              'splitter': splitter,
              'max_features': max_features,
              'max_depth': max_depth,
              'min_samples_split': min_samples_split,
              'min_samples_leaf': min_samples_leaf}
```

Para cada uno de los modelos se ejecuta el mismo tipo de entrenamiento con RandomizedSearchCV, pero con sus respectivos conjuntos de entrenamiento y validación:

```
Fitting 3 folds for each of 100 candidates, totalling 300 fits
RandomizedSearchCV
RandomizedSearchCV(cv=3, estimator=DecisionTreeRegressor(random_state=42),
                   n_iter=100, n_jobs=-1,
                   param_distributions={'criterion': ['squared_error',
                                                     'absolute_error',
                                                     'friedman_mse',
                                                     'poisson'],
                                         'max_depth': [10, 20, 30, 40, 50, 60,
                                                       70, 80, 90, 100, 110,
                                                       None],
                                         'max_features': ['auto', 'sqrt',
                                                          'log2'],
                                         'min_samples_leaf': [1, 2, 4],
                                         'min_samples_split': [2, 5, 10]},
                   ...)
```

4.7.1 Entrenamiento con todas las variables

Obteniendo los siguientes mejores parámetros:

```
{'splitter': 'best',
 'min_samples_split': 10,
 'min_samples_leaf': 2,
 'max_features': 'sqrt',
 'max_depth': 40,
 'criterion': 'absolute_error'}
```

Y los siguientes resultados:

```
MSE: 21.281676729971384  
MAE: 3.325752260504202  
MAPE: 0.05178672920322306
```

```
best_dt_rs1.score(x_train,y_train)
```

```
0.9823804261175073
```

4.7.2 Entrenamiento con variables seleccionadas

Obteniendo los siguientes mejores parámetros:

```
{'splitter': 'best',  
 'min_samples_split': 5,  
 'min_samples_leaf': 1,  
 'max_features': 'log2',  
 'max_depth': None,  
 'criterion': 'friedman_mse'}
```

Y los siguientes resultados:

```
MSE: 18.837880423891136  
MAE: 3.316825556022409  
MAPE: 0.05048329527779388
```

```
best_dt_rs2.score(x_train_selected,y_train)
```

```
0.9980907609010793
```

4.7.3 Entrenamiento con componentes principales

Obteniendo los siguientes mejores parámetros:

```
{'splitter': 'random',  
 'min_samples_split': 2,  
 'min_samples_leaf': 4,  
 'max_features': 'sqrt',  
 'max_depth': 10,  
 'criterion': 'absolute_error'}
```

Y los siguientes resultados:

```
MSE: 18.520603402461294  
MAE: 3.409973050420167  
MAPE: 0.0524140903835386
```

```
best_dt_rs3.score(x_train_pca,y_train)
```

```
0.8345324792763147
```

4.8 Random Forest

Se fija la siguiente malla de hiperparámetros:

```
rf = RandomForestRegressor(random_state = semilla)

n_estimators = [int(x) for x in np.linspace(start = 100, stop = 300, num = 10)]
criterion = ['squared_error', 'absolute_error', 'friedman_mse', 'poisson']
max_features = ['auto', 'sqrt','log2']
max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
max_depth.append(None)
min_samples_split = [2, 5, 10]
min_samples_leaf = [1, 2, 4]
bootstrap = [True, False]

param_grid = {'n_estimators': n_estimators,
              'criterion': criterion,
              'max_features': max_features,
              'max_depth': max_depth,
              'min_samples_split': min_samples_split,
              'min_samples_leaf': min_samples_leaf,
              'bootstrap': bootstrap}
```

Para cada uno de los modelos se ejecuta el mismo tipo de entrenamiento con RandomizedSearchCV, pero con sus respectivos conjuntos de entrenamiento y validación:

```
Fitting 3 folds for each of 100 candidates, totalling 300 fits
RandomizedSearchCV
RandomizedSearchCV(cv=3, estimator=RandomForestRegressor(random_state=42),
                   n_iter=100, n_jobs=-1,
                   param_distributions={'bootstrap': [True, False],
                                         'criterion': ['squared_error',
                                                       'absolute_error',
                                                       'friedman_mse',
                                                       'poisson'],
                                         'max_depth': [10, 20, 30, 40, 50, 60,
                                                       70, 80, 90, 100, 110,
                                                       None],
                                         'max_features': ['auto', 'sqrt',
                                                       'log2']},
                   ...)
```

4.8.1 Entrenamiento con todas las variables

Obteniendo los siguientes mejores parámetros:

```
{'n_estimators': 144,  
 'min_samples_split': 5,  
 'min_samples_leaf': 1,  
 'max_features': 'log2',  
 'max_depth': 60,  
 'criterion': 'squared_error',  
 'bootstrap': False}
```

Y los siguientes resultados:

```
MSE: 9.371245996132622  
MAE: 2.307437007489104  
MAPE: 0.03640346545498338
```

```
best_rf_rs1.score(x_train,y_train)
```

```
0.9996595823608789
```

4.8.2 Entrenamiento con variables seleccionadas

Obteniendo los siguientes mejores parámetros:

```
{'n_estimators': 233,  
 'min_samples_split': 5,  
 'min_samples_leaf': 1,  
 'max_features': 'sqrt',  
 'max_depth': 50,  
 'criterion': 'absolute_error',  
 'bootstrap': True}
```

Y los siguientes resultados:

```
MSE: 8.889034137439598  
MAE: 2.255379431600965  
MAPE: 0.03562418434241101
```

```
best_rf_rs2.score(x_train_selected,y_train)
```

```
0.9968534994082578
```

4.8.3 Entrenamiento con componentes principales

Obteniendo los siguientes mejores parámetros:

```
{'n_estimators': 255,
 'min_samples_split': 2,
 'min_samples_leaf': 1,
 'max_features': 'log2',
 'max_depth': 20,
 'criterion': 'squared_error',
 'bootstrap': False}
```

Y los siguientes resultados:

```
MSE: 10.623086912994376
MAE: 2.6392453638371998
MAPE: 0.040332475448647064
```

```
best_rf_rs3.score(x_train_pca,y_train)
```

```
0.9999978039266059
```

4.9 Gradient Boosting

Se fija la siguiente malla de hiperparámetros:

```
gb = GradientBoostingRegressor(random_state = semilla)

n_estimators = [int(x) for x in np.linspace(start = 100, stop = 1000, num = 10)]
criterion = ['squared_error', 'friedman_mse']
loss = ['squared_error', 'absolute_error', 'huber', 'quantile']
learning_rate = [0.01, 0.03, 0.05, 0.07, 0.09, 0.1]
subsample = [0.8,0.9,1.0]
max_features = ['auto', 'sqrt','log2']
max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
max_depth.append(None)
min_samples_split = [2, 5, 10]
min_samples_leaf = [1, 2, 4]
param_grid = {'n_estimators': n_estimators,
              'criterion': criterion,
              'loss': loss,
              'learning_rate': learning_rate,
              'subsample': subsample,
              'max_features': max_features,
              'max_depth': max_depth,
              'min_samples_split': min_samples_split,
              'min_samples_leaf': min_samples_leaf}
```

Para cada uno de los modelos se ejecuta el mismo tipo de entrenamiento con RandomizedSearchCV, pero con sus respectivos conjuntos de entrenamiento y validación:

```
Fitting 3 folds for each of 100 candidates, totalling 300 fits
[...]
RandomizedSearchCV
RandomizedSearchCV(cv=3, estimator=GradientBoostingRegressor(random_state=42),
   n_iter=100, n_jobs=-1,
   param_distributions={'criterion': ['squared_error',
                                      'friedman_mse'],
                        'learning_rate': [0.01, 0.03, 0.05,
                                          0.07, 0.09, 0.1],
                        'loss': ['squared_error',
                                 'absolute_error', 'huber',
                                 'quantile'],
                        'max_depth': [10, 20, 30, 40, 50, 60,
                                      70, 80, 90, 100, 110,
                                      ...]}
   [...]
   ► estimator: GradientBoostingRegressor
      ► GradientBoostingRegressor
```

4.9.1 Entrenamiento con todas las variables

Obteniendo los siguientes mejores parámetros:

```
{'subsample': 0.9,
 'n_estimators': 100,
 'min_samples_split': 5,
 'min_samples_leaf': 4,
 'max_features': 'sqrt',
 'max_depth': 80,
 'loss': 'squared_error',
 'learning_rate': 0.1,
 'criterion': 'squared_error'}
```

Y los siguientes resultados:

```
MSE: 9.90546717157191
MAE: 2.3335315715772245
MAPE: 0.036832479482317604
```

```
best_gb_rs1.score(x_train,y_train)
```

```
0.999904389262503
```

4.9.2 Entrenamiento con variables seleccionadas

Obteniendo los siguientes mejores parámetros:

```
{'subsample': 1.0,  
 'n_estimators': 1000,  
 'min_samples_split': 10,  
 'min_samples_leaf': 4,  
 'max_features': 'log2',  
 'max_depth': 40,  
 'loss': 'squared_error',  
 'learning_rate': 0.07,  
 'criterion': 'squared_error'}
```

Y los siguientes resultados:

```
MSE: 9.525007382598494  
MAE: 2.296507824553688  
MAPE: 0.036192027425133576
```

```
best_gb_rs2.score(x_train_selected,y_train)
```

```
0.9999999999992213
```

4.9.3 Entrenamiento con componentes principales

Obteniendo los siguientes mejores parámetros:

```
{'subsample': 1.0,  
 'n_estimators': 600,  
 'min_samples_split': 10,  
 'min_samples_leaf': 2,  
 'max_features': 'sqrt',  
 'max_depth': None,  
 'loss': 'quantile',  
 'learning_rate': 0.09,  
 'criterion': 'squared_error'}
```

Y los siguientes resultados:

```
MSE: 13.54012193009956  
MAE: 2.9717852847426647  
MAPE: 0.04630818958288136
```

```
best_gb_rs3.score(x_train_pca,y_train)
```

```
0.9562521655798545
```

4.10 Selección del mejor modelo

Según las métricas obtenidas, se aprecia, en su mayoría, una mejora en el desempeño de los modelos al utilizar el conjunto de entrenamiento con las variables seleccionadas. Por tal motivo, para evaluar el modelo final, este se entrenará con todo el conjunto de entrenamiento y validación que solo contienen las variables seleccionadas.

Los modelos obtenidos a partir de los 6 componentes principales tienen un peor desempeño que los demás modelos. Claramente, porque los componentes principales utilizados solo describen el 70% de la varianza de los datos. Sin embargo, es importante considerar que este 70% es descrito únicamente por 6 componentes, reduciendo la dimensionalidad de 23 a 6, ósea, una reducción de aproximadamente el 74% en la dimensionalidad del conjunto.

Entre los modelos entrenados, los que más destacan son el segundo Random Forest y el segundo Gradient Boosting. El Random Forest tiene mejores MSE, MAE y MAPE, pero un peor score.

Por su parte, el score obtenido para Gradient Boosting fue de aproximadamente 1. Esta métrica indica el coeficiente de determinación R^2 , el cual mide la proporción de la varianza en la variable dependiente que es explicada por el modelo. Un R^2 cercano a 1 indica un modelo que se ajusta muy bien a los datos. Sin embargo, un score tan alto es indicador de que ocurrió un overfitting en el conjunto de entrenamiento.

Tomando todo esto en consideración, el mejor modelo para el objetivo planteado es el Random Forest entrenado con el conjunto de variables seleccionadas en el backward selection.

4.11 Clasificación supervisada

Para el entrenamiento de los modelos de regresión logística multinomial no se utilizará un conjunto de validación, puesto que este modelo es únicamente exploratorio, y, debido a sus limitaciones inherentes en este caso de uso, no será empleado como modelo final. Por ello, se procede a combinar los conjuntos de entrenamiento y validación en un solo conjunto de entrenamiento para clasificación.

La limitación a la que se hace referencia es que el modelo está limitado a los grupos con los que será entrenado. Por tanto, si se entrena con grupos de intervalos de 2 años (60-62, 62-64, etc.) para que prediga, según las características del país, a qué grupo pertenecerá, la esperanza de vida máxima que podrá predecir será el último grupo definido. Por ejemplo, si el último grupo definido es el que corresponde a una esperanza de vida de 86-88, entonces el modelo no podrá hacer predicciones superiores a esta. Y, considerando la tendencia creciente en la esperanza de vida, se espera que el grupo máximo aumente en consecuencia.

```
[ ] x_train_classification = pd.concat([x_train, x_val])
x_train_classification_selected = pd.concat([x_train_selected, x_val_selected])
x_train_classification_pca = pd.concat([x_train_pca, x_val_pca])

y_train_classification_unprocessed = pd.concat([y_train, y_val])
```

Considerando que el mejor modelo de regresión obtenido previamente (Random Forest entrenado con las variables seleccionadas) tuvo un MAE de 2.25 años, se evaluará si un modelo de regresión logística multinomial puede equiparar esta precisión clasificando los países por grupos con intervalos de 2 años (60-62, 62-64, etc.).

Para esto, se toma el valor mínimo de esperanza de vida del dataset y se generan grupos con rangos de 2 años hasta el valor máximo de esperanza de vida.

```
[ ] bins = np.arange(36, 88, 2)
labels = np.arange(1, len(bins))

y_train_classification = pd.cut(y_train_classification_unprocessed, bins=bins, labels=labels, right=False).astype(int)
y_test_classification = pd.cut(y_test, bins=bins, labels=labels, right=False).astype(int)
```

4.11.1 Regresión Logística Multinomial

Se fija la siguiente malla de hiperparámetros:

```
[ ] lr = LogisticRegression(multi_class='multinomial', max_iter=1000, random_state=semilla)

param_grid = {
    'C': loguniform(1e-4, 1e2),
    'penalty': ['l1', 'l2', 'elasticnet', None],
    'fit_intercept': [True, False],
    'tol': [1e-4, 1e-3, 1e-2],
    'class_weight': [None, 'balanced'],
    'solver': ['lbfgs', 'liblinear', 'newton-cg', 'newton-cholesky', 'sag', 'saga']
}
```

Para cada uno de los modelos se ejecuta el mismo tipo de entrenamiento con RandomizedSearchCV, pero con sus respectivos conjuntos de entrenamiento y prueba:

```
Fitting 3 folds for each of 100 candidates, totalling 300 fits
RandomizedSearchCV
RandomizedSearchCV(cv=3,
    estimator=LogisticRegression(max_iter=1000,
        multi_class='multinomial',
        random_state=42),
    n_iter=100, n_jobs=-1,
    param_distributions={'C': <scipy.stats._distn_infrastructure.rv_continuous_frozen object at 0x7ce0afbe9190>,
        'class_weight': [None, 'balanced'],
        'fit_intercept': [True, False],
        'penalty': ['l1', 'l2', 'elasticnet',
            None],
        'solver': ['lbfgs', 'liblinear',
            ...]}

    ▶ best_estimator_:
    LogisticRegression
```

4.11.1.1 Entrenamiento con todas las variables

Obteniendo los siguientes mejores parámetros:

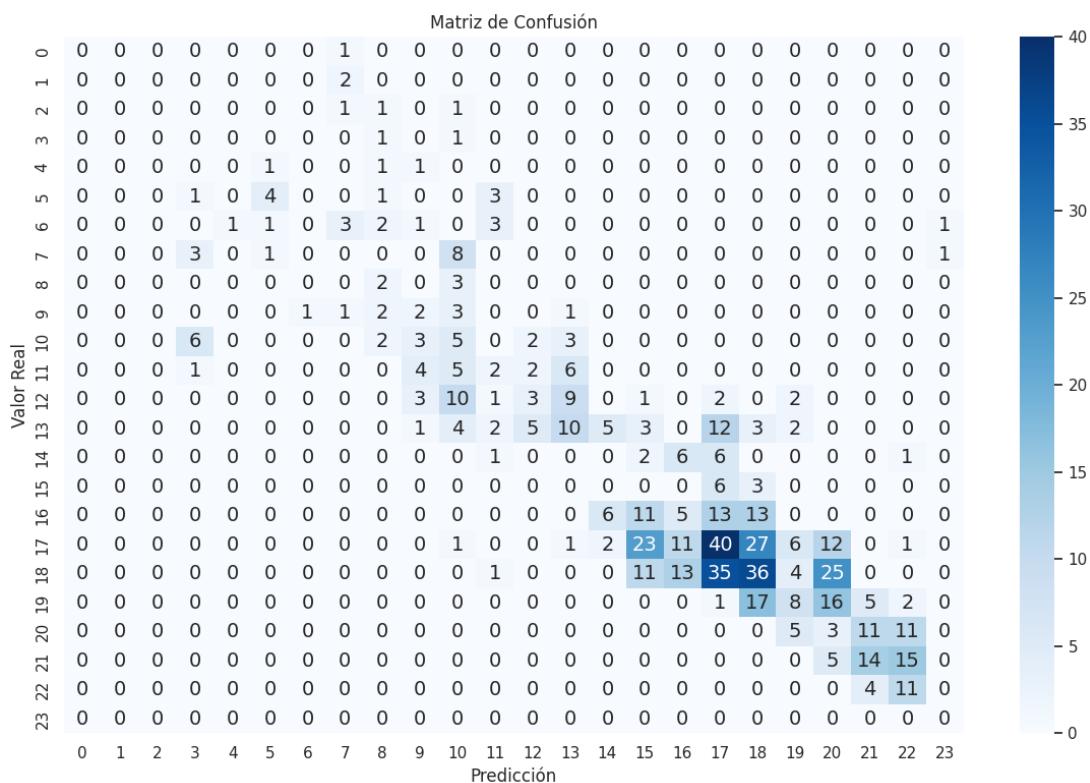
```
{'C': np.float64(0.7115245297181694),
 'class_weight': None,
 'fit_intercept': True,
 'penalty': None,
 'solver': 'newton-cholesky',
 'tol': 0.001}
```

El siguiente reporte de clasificación:

Reporte de clasificación:					
	precision	recall	f1-score	support	
2	0.00	0.00	0.00	1	
3	0.00	0.00	0.00	2	
4	0.00	0.00	0.00	3	
5	0.00	0.00	0.00	2	
6	0.00	0.00	0.00	3	
7	0.57	0.44	0.50	9	
8	0.00	0.00	0.00	12	
9	0.00	0.00	0.00	13	
10	0.17	0.40	0.24	5	
11	0.13	0.20	0.16	10	
12	0.12	0.24	0.16	21	
13	0.15	0.10	0.12	20	
14	0.25	0.10	0.14	31	
15	0.33	0.21	0.26	47	
16	0.00	0.00	0.00	16	
17	0.00	0.00	0.00	9	
18	0.14	0.10	0.12	48	
19	0.35	0.32	0.33	124	
20	0.36	0.29	0.32	125	
21	0.30	0.16	0.21	49	
22	0.05	0.10	0.07	30	
23	0.41	0.41	0.41	34	
24	0.27	0.73	0.39	15	
25	0.00	0.00	0.00	0	
accuracy				0.23	629
macro avg		0.15	0.16	0.14	629
weighted avg		0.26	0.23	0.24	629

Accuracy: 0.2305

Y la siguiente matriz de confusión:



4.11.1.2 Entrenamiento con variables seleccionadas

Obteniendo los siguientes mejores parámetros:

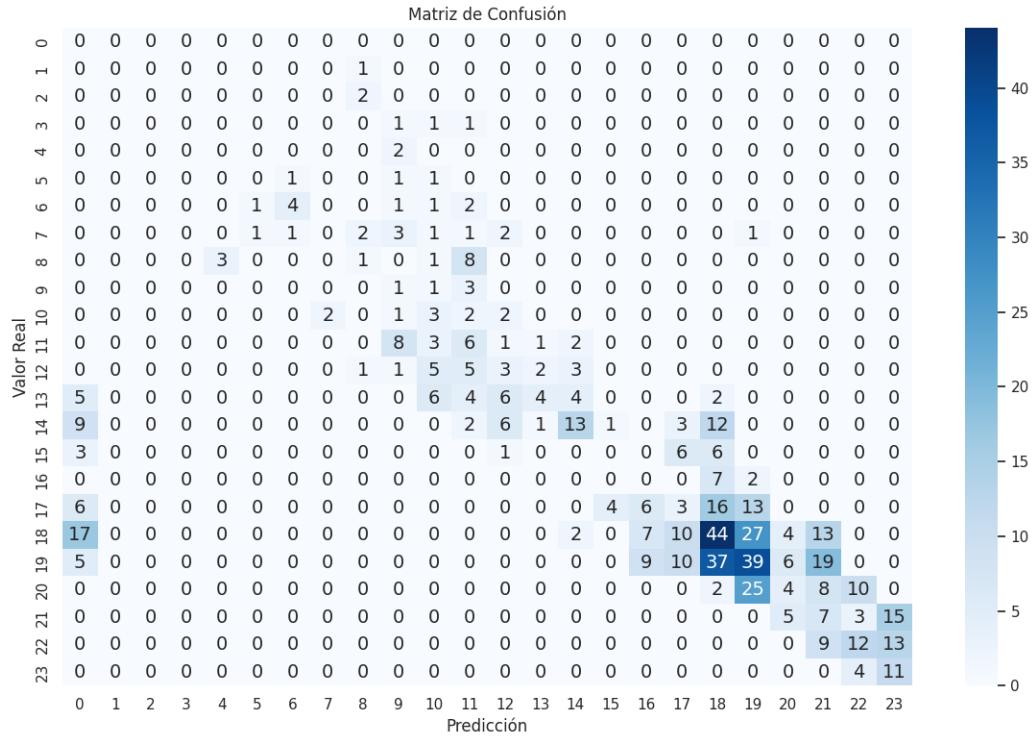
```
{'C': np.float64(0.6326714903289286),  
 'class_weight': None,  
 'fit_intercept': True,  
 'penalty': None,  
 'solver': 'lbfgs',  
 'tol': 0.0001}
```

El siguiente reporte de clasificación:

Reporte de clasificación:				
	precision	recall	f1-score	support
1	0.00	0.00	0.00	0
2	0.00	0.00	0.00	1
3	0.00	0.00	0.00	2
4	0.00	0.00	0.00	3
5	0.00	0.00	0.00	2
6	0.00	0.00	0.00	3
7	0.67	0.44	0.53	9
8	0.00	0.00	0.00	12
9	0.14	0.08	0.10	13
10	0.05	0.20	0.08	5
11	0.13	0.30	0.18	10
12	0.18	0.29	0.22	21
13	0.14	0.15	0.15	20
14	0.50	0.13	0.21	31
15	0.54	0.28	0.37	47
16	0.00	0.00	0.00	16
17	0.00	0.00	0.00	9
18	0.09	0.06	0.07	48
19	0.35	0.35	0.35	124
20	0.36	0.31	0.34	125
21	0.21	0.08	0.12	49
22	0.12	0.23	0.16	30
23	0.41	0.35	0.38	34
24	0.28	0.73	0.41	15
accuracy				0.25
macro avg				0.17
weighted avg				0.29

Accuracy: 0.2464

Y la siguiente matriz de confusión:



4.11.1.3 Entrenamiento con componentes principales

Obteniendo los siguientes mejores parámetros:

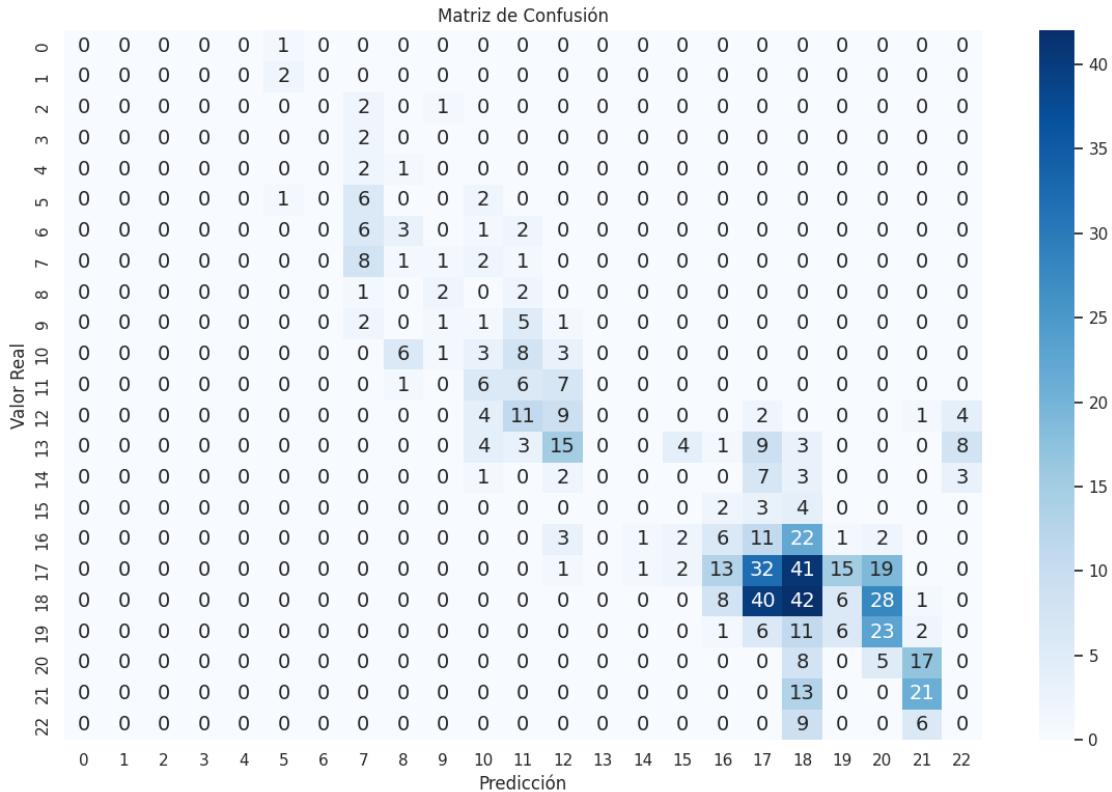
```
{'C': np.float64(0.9145011591835932),
 'class_weight': None,
 'fit_intercept': True,
 'penalty': None,
 'solver': 'sag',
 'tol': 0.01}
```

El siguiente reporte de clasificación:

Reporte de clasificación:					
	precision	recall	f1-score	support	
2	0.00	0.00	0.00		1
3	0.00	0.00	0.00		2
4	0.00	0.00	0.00		3
5	0.00	0.00	0.00		2
6	0.00	0.00	0.00		3
7	0.25	0.11	0.15		9
8	0.00	0.00	0.00		12
9	0.28	0.62	0.38		13
10	0.00	0.00	0.00		5
11	0.17	0.10	0.12		10
12	0.12	0.14	0.13		21
13	0.16	0.30	0.21		20
14	0.22	0.29	0.25		31
15	0.00	0.00	0.00		47
16	0.00	0.00	0.00		16
17	0.00	0.00	0.00		9
18	0.19	0.12	0.15		48
19	0.29	0.26	0.27		124
20	0.27	0.34	0.30		125
21	0.21	0.12	0.16		49
22	0.06	0.17	0.09		30
23	0.44	0.62	0.51		34
24	0.00	0.00	0.00		15
accuracy				0.22	629
macro avg		0.12	0.14	0.12	629
weighted avg		0.20	0.22	0.20	629

Accuracy: 0.2226

Y la siguiente matriz de confusión:



4.11.2 Selección del mejor modelo de clasificación supervisada

Con base a los resultados obtenidos, se determina que el mejor modelo de clasificación supervisada es el de regresión logística multinomial entrenado sobre el conjunto de variables seleccionadas. El modelo predice correctamente aproximadamente el 24.6% de los casos totales. En un problema multiclasa con 24 clases, esto no es particularmente alto, pero podría ser aceptable dependiendo del caso de uso que se le quiera dar.

Con base en los promedios macro: Precision = 0.17, Recall = 0.17, y F1-score = 0.15, se identifica que el rendimiento general por clase es bajo. Sin embargo, el promedio ponderado indica un mejor rendimiento, pero a la vez, confirma que el modelo está sesgado hacia las clases con más observaciones, un comportamiento que era de esperarse.

Las clases de mejor desempeño son las 7 (48-50 años), 15 (64-66 años), 19 (72-74 años), 20 (74-76 años), 23 (80-82 años) y 24 (82-84 años) con 0.53, 0.37, 0.35, 0.34, 0.38, y 0.41 de f1-score respectivamente. Las clases 19 y 20 son las que presentan mayor cantidad de muestras en el conjunto de prueba, con 124 y 125 respectivamente. Las clases 7, 15, 23 y 24 presentaban 9, 47, 34 y 15 muestras respectivamente.

Estas métricas indican que el modelo tiene buen desempeño en los grupos con mayor representación en el conjunto de prueba, y, en su mayoría, aquellos que corresponden a esperanzas de vida medias o altas. No obstante, debido a la cercanía de los grupos, el modelo no se desempeña tan bien en algunas clases como las 16, 17, 18, 21 y 22, puesto que el modelo puede confundirse con otras clases cercanas a estas.

Además de estas clases de bajo desempeño como la 16 y 17, las del 1 al 6 (de 36 a 48 años), y del 8 al 10 (50 a 56 años), presentaron f1-score muy bajos o nulos. Esto se debe en mayor medida a la falta de muestras y al sesgo a favor de las clases superiores, lo que confirma que el modelo esta entrenado a predecir con mejor precisión los países con mayor esperanza de vida.

Según la matriz de confusión, el modelo tiende a predecir clases adyacentes o similares para las verdaderas. Por ejemplo:

- Clase real 18 y 20 se predice frecuentemente como 19.
- Clase real 19 se confunde con 18.
- Clase 22 se confunde con 21 y 23.
- Clase 13 se confunde con 14, 12, 11 y 10.

Esto indica que las características utilizadas no son lo suficientemente discriminantes para separar ciertas clases. Idealmente, la matriz debería tener valores altos solo en la diagonal (predicción correcta), pero aquí hay muchas entradas fuera de la diagonal (diagonal difusa), indicando confusión frecuente entre clases. En particular, las clases del 18 al 24 tienen mayor confusión entre sí, probablemente porque comparten características similares.

4.12 Clasificación no supervisada

Para la clasificación no supervisada con KMeans, solo se les aplicará a los datos de 2016. Para ello, se deben volver a procesar los conjuntos de entrenamiento y prueba, siguiendo los pasos a continuación:

- Filtrar la variable "year" para el valor 2016
- Estandarizar los datos

- Convertir las variables categóricas a numéricas
- Eliminar las variables innecesarias, incluyendo "life_expect" por ser un problema no supervisado

Para determinar en cuantos clusters se deberían segmentar los datos para el KMeans, se debe hacer una exploración estadística de la variable "life_expect", evaluando su distribución y estadísticas descriptivas.

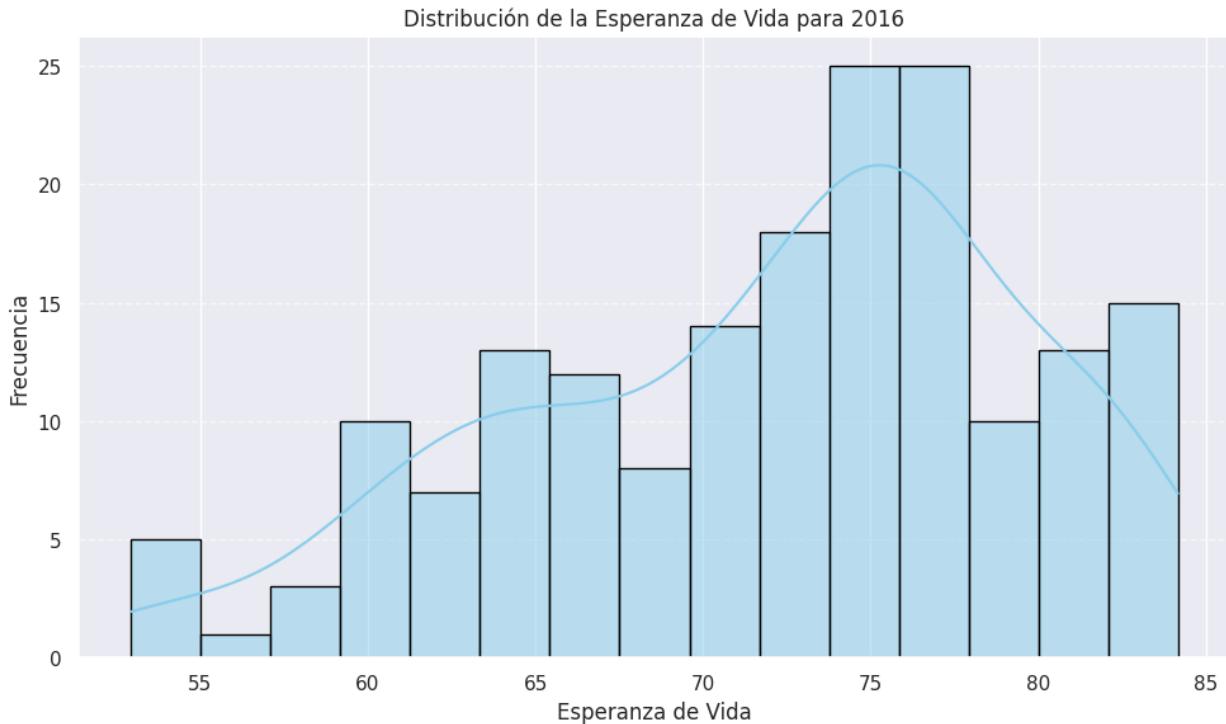
Media: 71.99
 Varianza: 56.75
 Desviación estándar: 7.53
 Mínimo: 52.94
 Máximo: 84.17

- La media indica que, en promedio, la esperanza de vida de los países del dataset es aproximadamente **72 años**.
- La varianza mide la dispersión de los datos respecto a la media. Un valor de **56.75** sugiere una dispersión moderada: hay países con esperanza de vida significativamente más baja o alta que el promedio.
- Una desviación estándar de **7.53** indica que la mayoría de los valores se encuentran dentro del rango: $[72 - 7.5, 72 + 7.5]$ lo que es igual a **[64.5, 79.5]**. Esto sugiere que, aunque hay dispersión, no hay una heterogeneidad extrema.
- Una esperanza de vida mínima de **52.94** años y una máxima de **84.17** años.



Del boxplot se identifican 3 grandes zonas visto de manera muy general:

- Países con esperanza de vida baja: (**52.94 - ~66 años**).
- Países con esperanza de vida media: (**~66 - ~77 años**)
- Países con esperanza de vida alta: (**~77 - 84.17 años**)



El histograma indica una distribución sesgada a la izquierda. Esta evidencia la existencia de 3 picos, lo que sugiere tres grupos naturales en correspondencia con lo evidenciado en el boxplot.

Con base a todo lo anterior, se determina que el número de clusters ideal para segmentar los datos son 3. Esto da mayor interpretabilidad al modelo esperado, puesto que se prevé que los clusters tendrán una distribución parecida a la siguiente:

- **Cluster 1:** Esperanza de vida baja (países con crisis sanitarias, conflictos, subdesarrollo)
- **Cluster 2:** Esperanza de vida media (países en transición económica o con servicios de salud en crecimiento)
- **Cluster 3:** Esperanza de vida alta (países desarrollados con sistemas de salud robustos)

De igual forma, usar más de 3 clusters podría forzar subdivisiones sin interpretación clara (por ejemplo, separar países con 75 vs 78 años de esperanza de vida). Por otra parte, menos de 3 clusters perdería información relevante al, por ejemplo, colapsar países con 55 años de esperanza vida con países de 72.

Se procede a construir el modelo de KMeans con 3 clusters. Igualmente, se clasifica la esperanza de vida real de los países en 3 clusters, los cuales se dividen según el siguiente criterio:

- **Cluster 1:** todos los valores menores a la media menos 1 desviación estándar.
- **Cluster 2:** todos los valores entre la media menos 1 desviación estándar y la media más 1 desviación estándar.
- **Cluster 3:** todos los valores mayores a la media más 1 desviación estándar.

Con base en esta clasificación, se comparan los clusters asignados por KMeans, con los clusters generados manualmente obteniendo el siguiente dataframe:

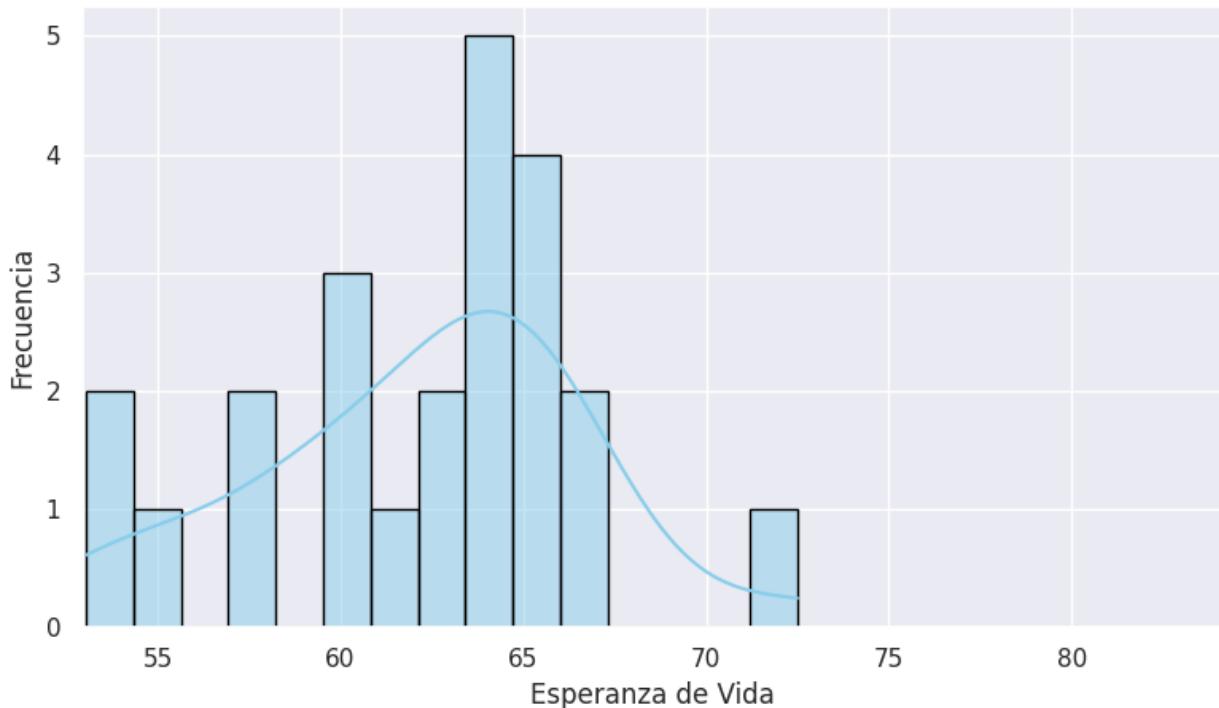
	region	country	life_expect	cluster_kmeans	cluster_life_expect	comparacion
0	Africa	Angola	62.63262	1	1	True
1	Africa	Burundi	60.09811	3	1	False
2	Africa	Benin	61.08568	1	1	True
3	Africa	Burkina Faso	60.32101	3	1	False
4	Africa	Botswana	66.05297	3	2	False
5	Africa	Central African Republic	53.04280	1	1	True
6	Africa	United Republic of Tanzania	63.90913	3	1	False
7	Africa	Uganda	62.49541	3	1	False
8	Africa	South Africa	63.59951	3	1	False
9	Africa	Zambia	62.32869	3	1	False

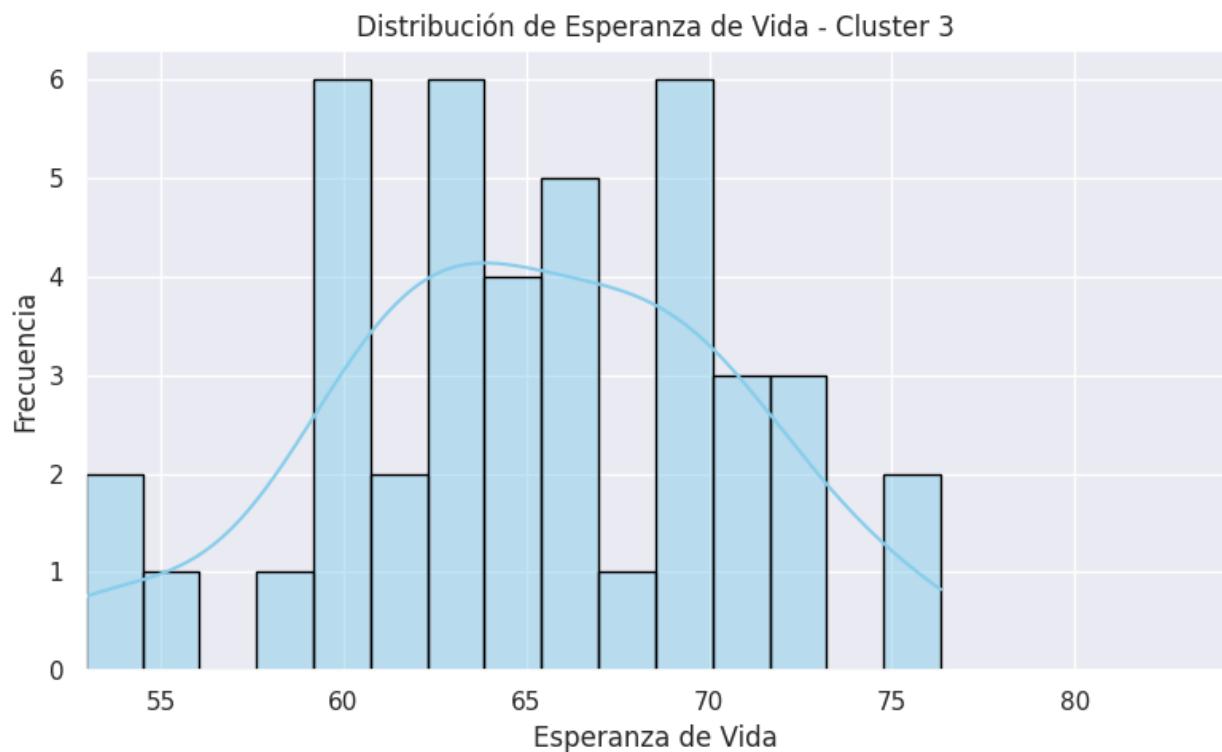
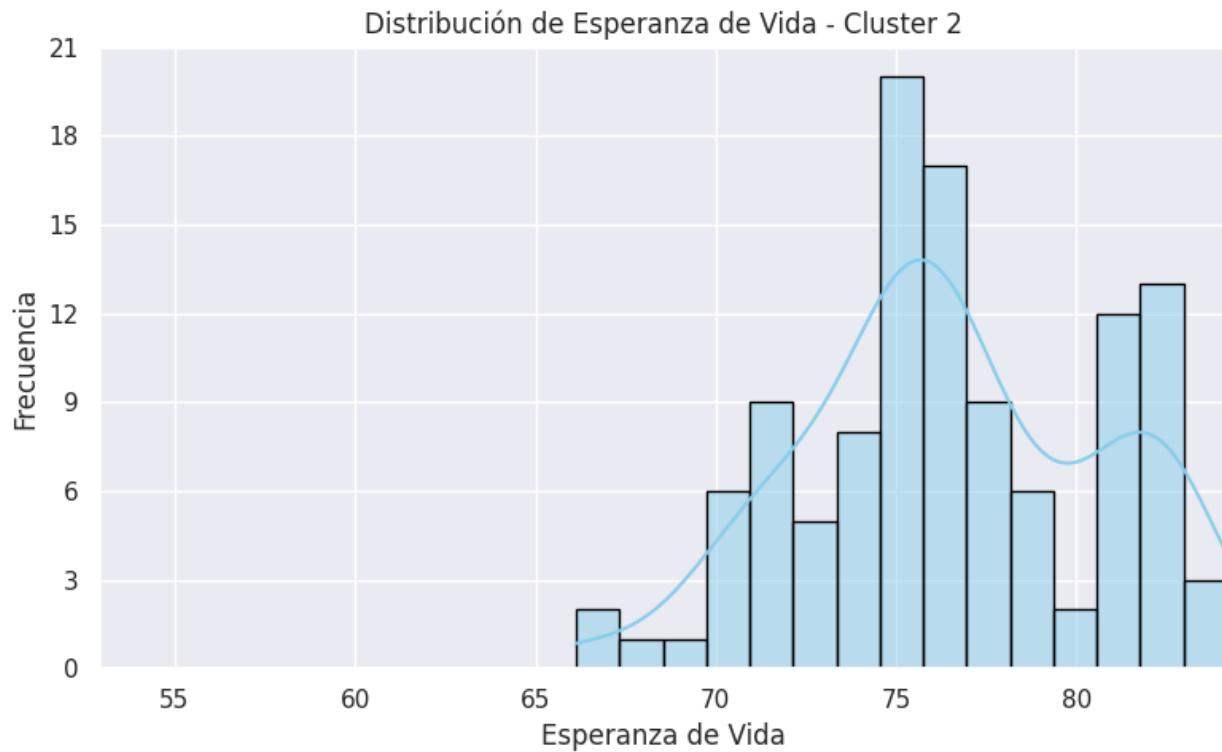
El promedio y la mediana por región es la siguiente:

	region	promedio	mediana
0	Africa	62.610828	62.564015
1	Americas	74.800919	75.173040
2	Eastern Mediterranean	72.293639	74.752180
3	Europe	77.819102	77.757440
4	South-East Asia	71.614260	70.387800
5	Western Pacific	74.134015	73.429950

La distribución de los clusters generados por KMeans es la siguiente:

Distribución de Esperanza de Vida - Cluster 1



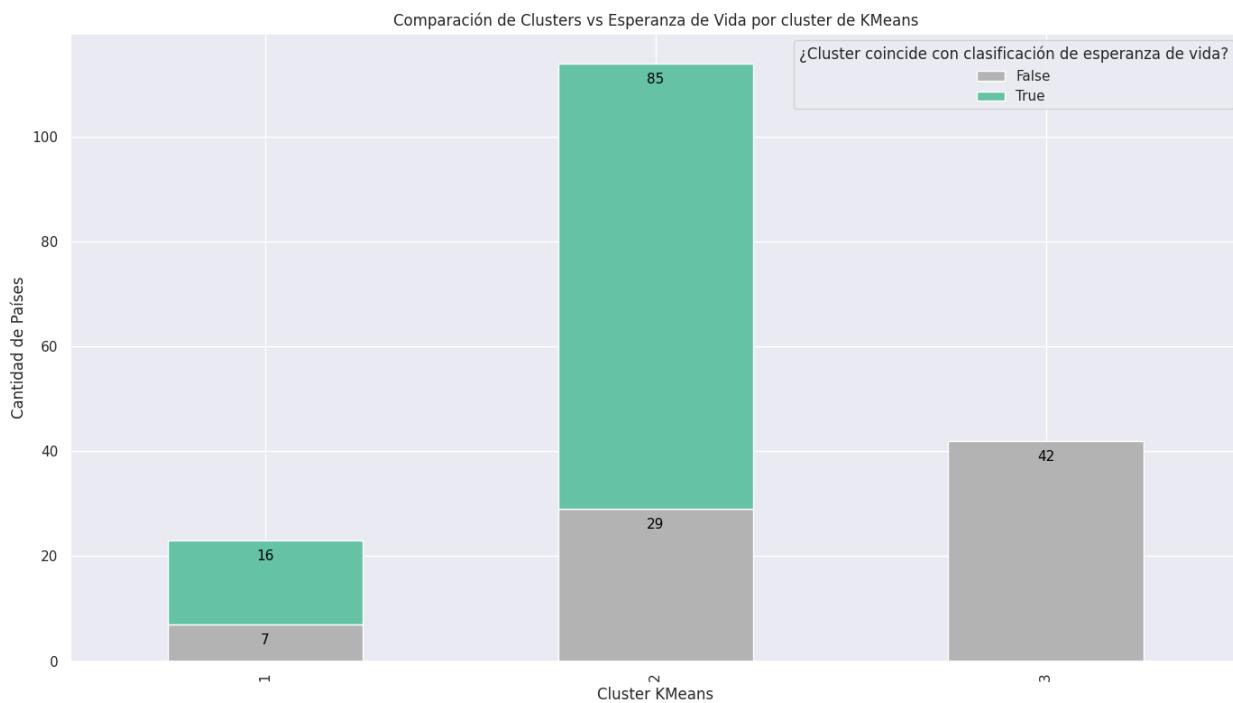


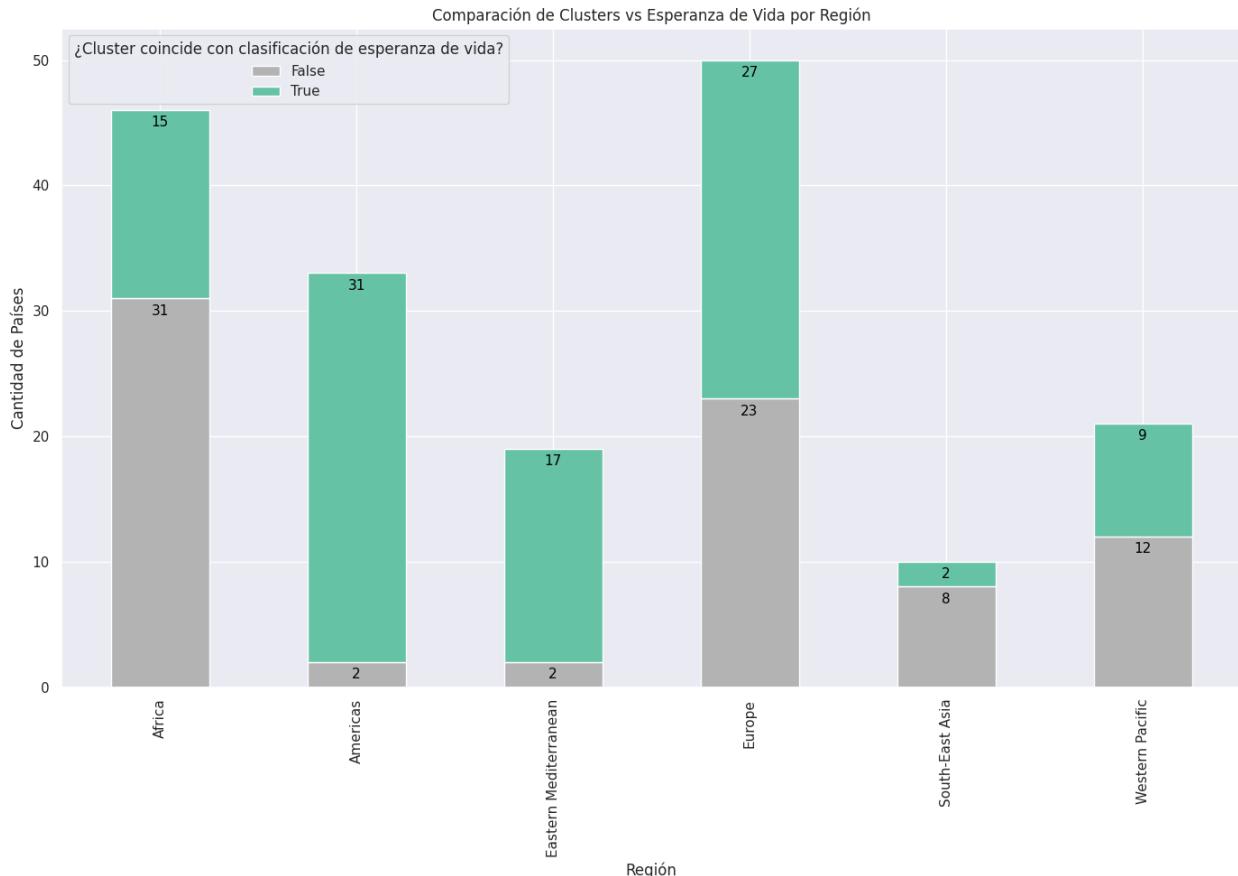
Estos clusters generados por KMeans muestran la siguiente distribución:

- **Cluster 1:** Agrupa los países con esperanzas de vida bajas y medias con mayor proporción en las bajas.

- **Cluster 2:** Agrupa los países con esperanzas de vida medias y altas con mayor proporción en las altas.
- **Cluster 3:** Agrupa los países con esperanzas de vida bajas y medias con mayor proporción en las medias.

De la comparación entre estos clusters y los generados manualmente con base a la esperanza de vida de los países, se obtienen los resultados a continuación:



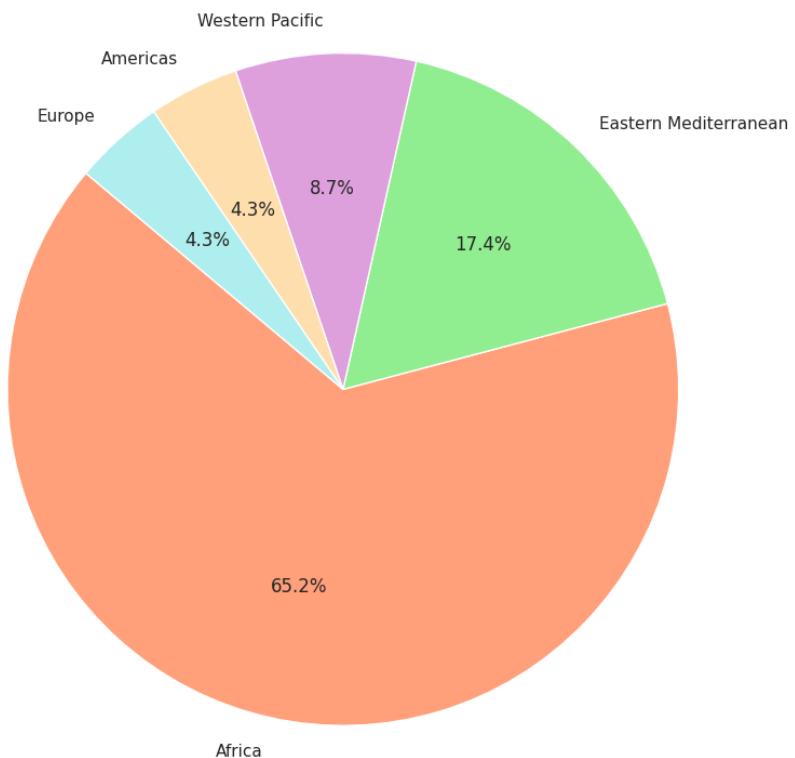


Con base en los gráficos, se pueden apreciar las siguientes tendencias en la segmentación generada por KMeans:

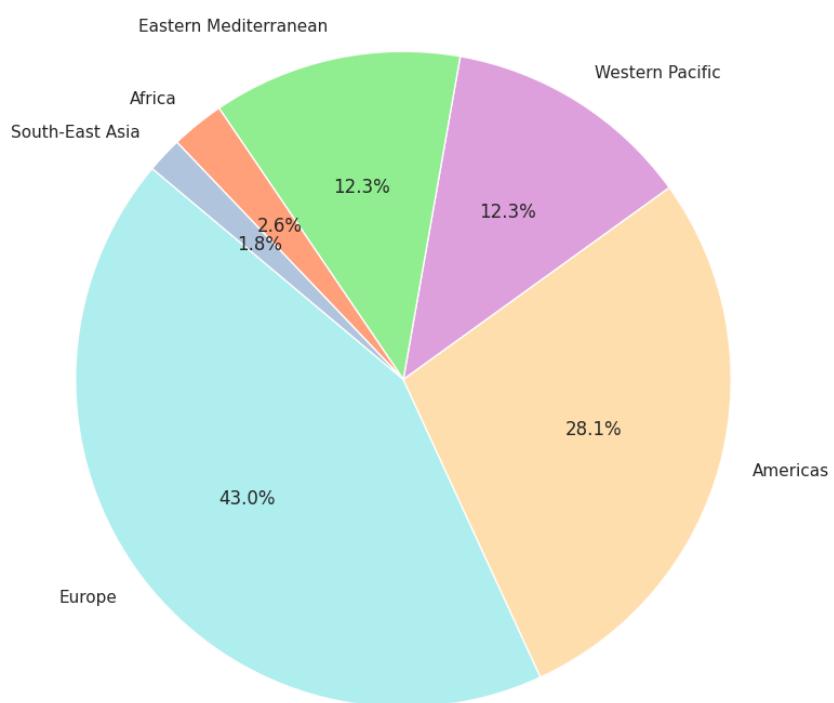
- 101 países fueron clasificados en correspondencia a la segmentación manual, mientras que 78 no lo fueron.
- Dentro de los 101 países clasificados en correspondencia a la segmentación manual, 85 fueron clasificados en el cluster 2, y 16 en el 1. Mientras que, de los 78 clasificados de manera diferente, 7 fueron clasificados en el cluster 1, 29 en el 2, y 42 en el 3. Se puede apreciar que la mayoría de los datos fueron clasificados en el cluster 2, el cual agrupa los países con esperanzas de vida medias y altas con mayor proporción en las altas.
- De las regiones que presentaron mayor coincidencia con la distribución generada manualmente, están "Americas", "Eastern Mediterranean", y "Europe" con un 94%, 89% y 54% de coincidencia. Las regiones con menor coincidencia fueron "Western Pacific", "Africa" y "South-East Asia" con 43%, 33% y 20%.

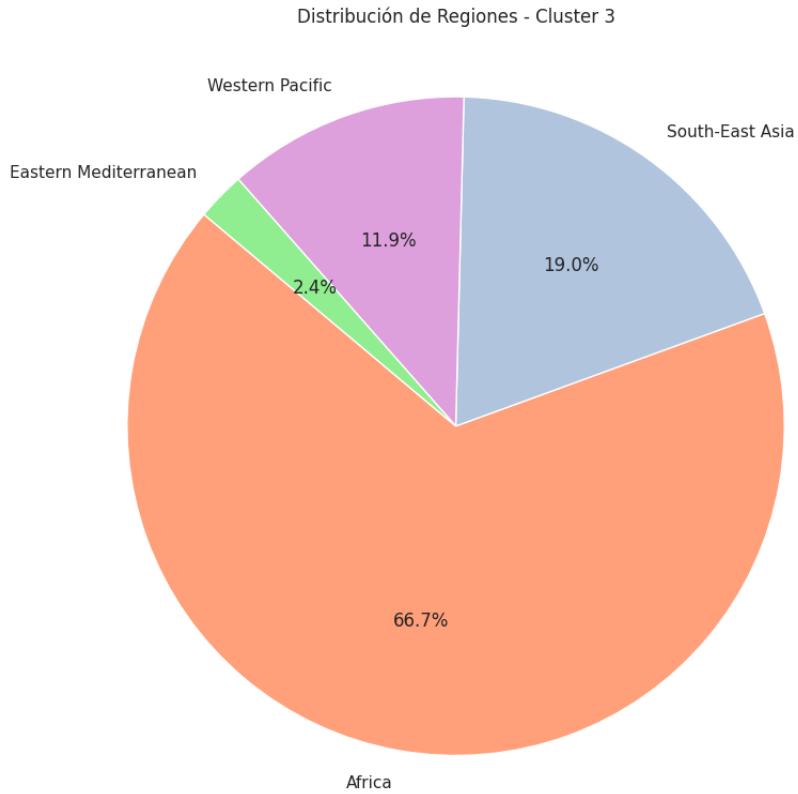
Por último, la composición de los clusters por región es la siguiente:

Distribución de Regiones - Cluster 1



Distribución de Regiones - Cluster 2





Estos gráficos de pastel indican las siguientes tendencias:

- **Cluster 1:** Compuesto en un 65.2% por países Africanos, seguido por un 17.4% por países del Mediterraneo Este. Esto tiene sentido, puesto que el cluster 1 agrupa los países de menor esperanza de vida, y la región de Africa tiene el menor promedio de esperanza de vida de todas las regiones, seguido por la región del Sudeste de Asia y la del Mediterraneo Este. Resulta extraña la presencia de las regiones Europa y Americas en este cluster, puesto que estas son las dos regiones de mayor promedio de esperanza de vida.
- **Cluster 2:** Compuesto en un 43% por países Europeos, seguido por 28.1% Americanos y 12.3% del Pacífico Oeste. Estas tres regiones son, a la vez, las de mayor promedio de esperanza de vida, por lo que tiene sentido que hagan parte del cluster 2, el cual agrupa los países de mayor esperanza de vida.
- **Cluster 3:** Tiene una composición similar a la del cluster 1, con un 66.7% países Africanos, seguido esta vez por un 19% de países del Sudeste de Asia. Ya que este cluster tiene una agrupación de esperanzas de vida similar al 1, se evidencia similaridad con la distribución de regiones del cluster 1.

5. Evaluación

5.1 Métricas MAE, MSE y MAPE

Ya escogido el modelo final, se procede a entrenarlo con el conjunto de entrenamiento y evaluación combinados.

```

x_train_val_selected = pd.concat([x_train_selected, x_val_selected])
y_train_val = pd.concat([y_train, y_val])

x_train_val_selected.shape
(2414, 18)

y_train_val.shape
(2414,)

final_model.fit(x_train_val_selected, y_train_val)

RandomForestRegressor(criterion='absolute_error', max_depth=50,
                      max_features='sqrt', min_samples_split=5,
                      n_estimators=233, random_state=42)

pred_test_fm = final_model.predict(x_test_selected)

print("MSE: " + str(mean_squared_error(y_test, pred_test_fm)))
print("MAE: " + str(mean_absolute_error(y_test, pred_test_fm)))
print("MAPE: " + str(mean_absolute_percentage_error(y_test, pred_test_fm)))

MSE: 9.240226629727047
MAE: 2.2841096852078038
MAPE: 0.034626150855223685

print("MSE: " + str(mean_squared_error(y_val, pred_val_best_rf_rs2)))
print("MAE: " + str(mean_absolute_error(y_val, pred_val_best_rf_rs2)))
print("MAPE: " + str(mean_absolute_percentage_error(y_val, pred_val_best_rf_rs2)))

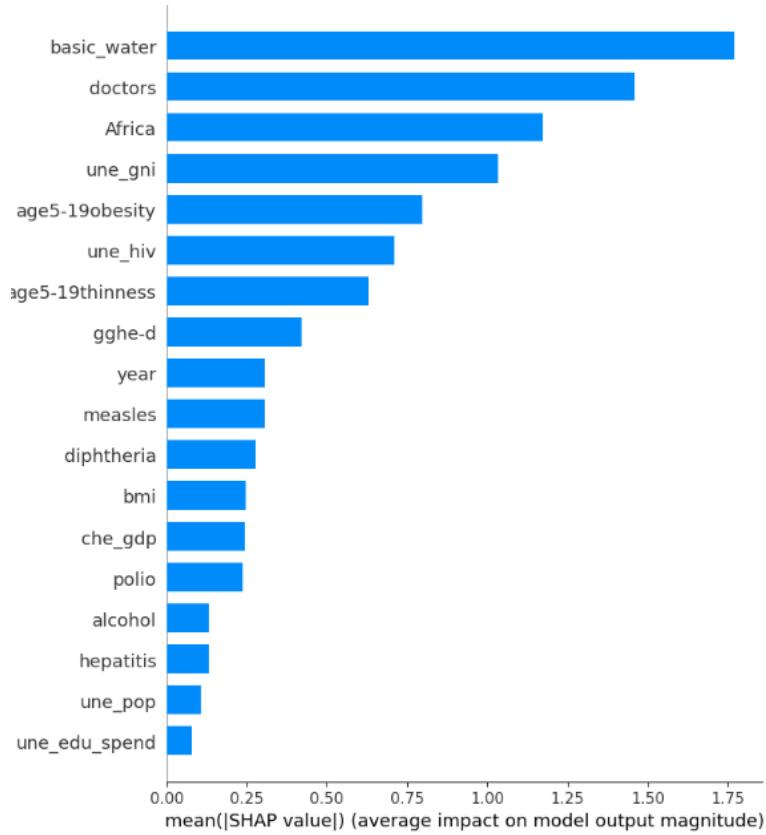
MSE: 8.889034137439598
MAE: 2.255379431600965
MAPE: 0.03562418434241101

```

Al comparar las métricas del conjunto de prueba con las del conjunto de validación, se observa que estas son muy cercanas entre sí, lo cual indica que el modelo generaliza bien y no está sobre ajustado a los datos de entrenamiento y validación.

5.2 Análisis de desempeño

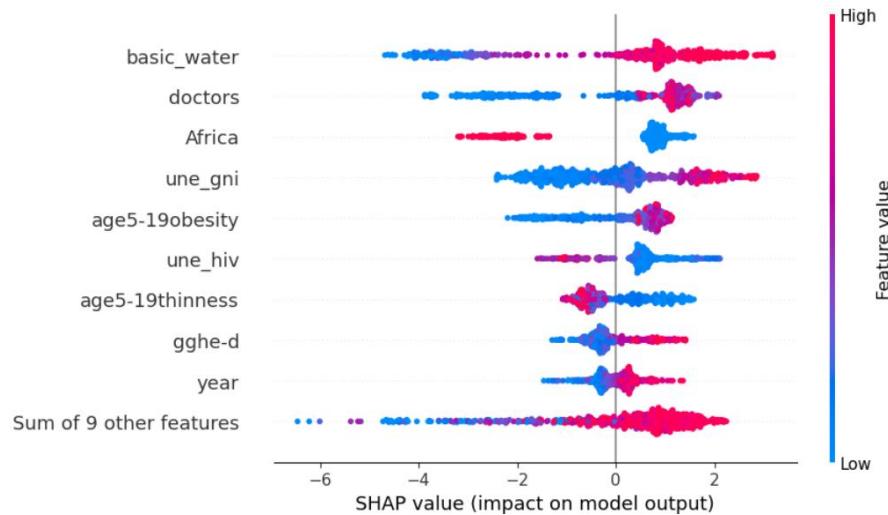
Para evaluar el desempeño del modelo final sobre el conjunto de prueba se verifica su comportamiento en distintos escenarios, analizando tanto sus resultados como sus errores.



Con base en la gráfica, se aprecia un decrecimiento gradual y pronunciado en la importancia de las variables, desde basic_water hasta year. A partir de year, el decrecimiento en importancia se torna menos significativo.

Las 5 variables con mayor impacto en la predicción del modelo fueron basic_water, doctors, Africa, une_gni, y age5-19obesity. Considerando los diferentes análisis realizados a lo largo del proyecto, cobra sentido que estas sean las variables de mayor importancia. No obstante, en el caso de doctors, al ser una de las características con mayor cantidad de datos faltantes, su elevado nivel de importancia representa un mayor grado de sesgo e imprecisión. Esto debido a que más de un tercio de la totalidad de sus valores fueron imputados.

Finalmente, a diferencia de los valores SHAP del modelo de Gradient Boosting utilizado en la sección 3, en este caso no existe una diferencia tan grande entre basic_water y la segunda variable más importante. En el primer modelo, la diferencia del impacto promedio en la predicción entre la primera y segunda variable era de aproximadamente 3.7. Para el modelo final evaluado en el conjunto de prueba, esta diferencia es de aproximadamente 0.3.



De este gráfico se evidencian tendencias marcadas que son coherentes con el comportamiento esperado del modelo. Igualmente, a simple vista, no se identifican muchos outliers.

Posterior a la evaluación de los resultados, se calculan todos los errores absolutos en el conjunto de prueba para estudiar los errores del modelo.

```
err_abs_fm = abs(pred_test_fm - y_test)
err_abs_fm
```

51	1.007020
52	1.954598
53	1.696415
54	1.620235
55	1.039817
...	
3038	1.357454
3039	1.919283
3040	2.563551
3041	2.005669
3042	0.055681

Name: life_expect, Length: 629, dtype: float64

Se comprueba que la media de los errores absolutos es el mismo MAE del modelo.

```
err_abs_fm.mean()
```

2.2841096852078038

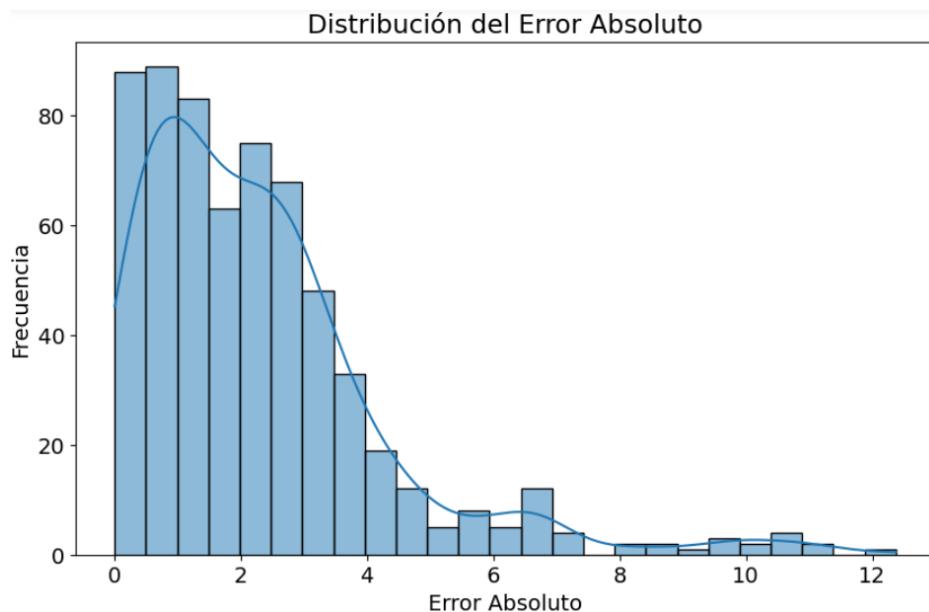
```
mean_absolute_error(y_test, pred_test_fm)
```

2.2841096852078038

Se agrega la columna de errores absolutos al dataframe data_imputed para poder realizar un estudio más detallado de los errores del modelo con cada variable en su respectiva escala original.

```
data_imputed_test = data_imputed[data_imputed.index.isin(x_test_selected.index)]
data_imputed_test["Error absoluto"] = err_abs_fm
data_imputed_test.head()
```

	che_gdp	une_edu_spend	basic_water	doctors	bmi	age5-19thinness	age5-19obesity	alcohol	hepatitis	measles	polio	diphtheria	une_hiv	life_expect	Error absoluto
1	3.31752	4.4326	54.91951	0.543	20.9	11.5	0.1	4.50590	76.0	48.0	45.0	45.0	2.1	50.08198	1.007020
2	3.24898	4.4326	55.08078	0.543	21.0	11.3	0.1	5.52657	76.0	63.0	62.0	62.0	2.0	50.46167	1.954598
3	3.34058	4.4326	54.57504	0.543	21.1	11.1	0.2	5.48323	76.0	64.0	69.0	69.0	1.8	50.90548	1.696415
4	3.53624	4.4326	54.08577	0.543	21.2	10.8	0.2	5.56515	76.0	76.0	83.0	79.0	1.7	51.41922	1.620235
5	4.74466	4.4326	53.61477	0.543	21.3	10.6	0.2	5.51264	76.0	78.0	83.0	79.0	1.6	52.11454	1.039817



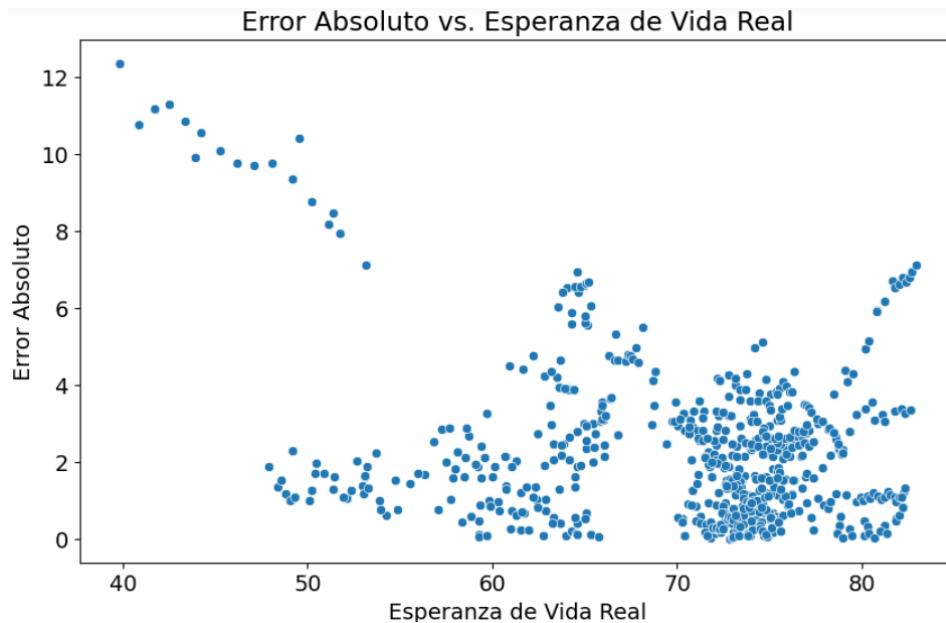
La distribución del error absoluto en el conjunto de prueba muestra la mayor concentración entre 0 y 4. En este rango, los dos picos de errores de la distribución, que son relativamente iguales, están entre 0.5 y 1 el mayor, y 0 y 0.5 el que le sigue. Esto significa entonces que, aunque el MAE es de 2.2841096852078038, la mayor concentración de errores absolutos está entre 0 y 1.5.

```
data_imputed_test["Error absoluto"].median()
```

1.8828193347639228

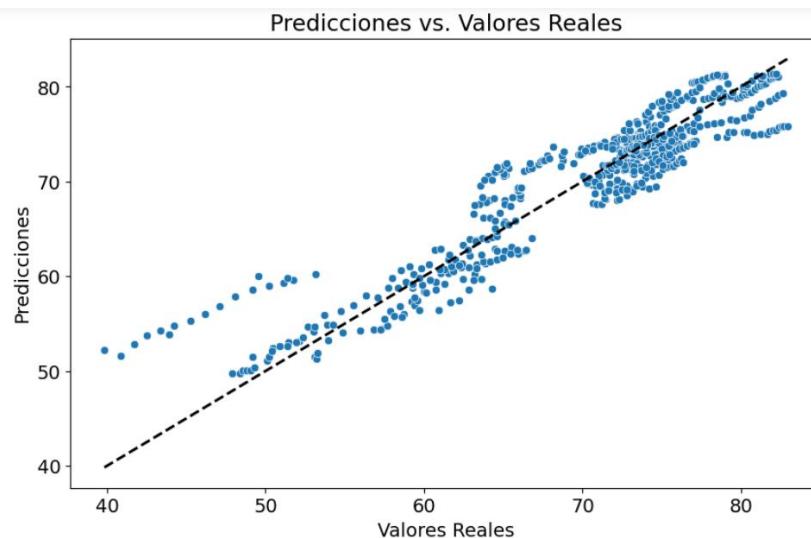
Como se puede observar, la mediana del error absoluto es de 1.8828193347639228, ósea, el 50% de los resultados tuvieron un error absoluto igual o menor a esta cifra. El MAE, por su parte, al no ser una métrica robusta, se ve afectada por los outliers en la distribución de errores absolutos.

Estos outliers probablemente pertenezcan a países con baja esperanza de vida, los cuales, en su mayoría, son de la región de África. Esta conjetura se basa tanto en las observaciones de outliers en la sección 3, como en la lógica de que estos países son considerablemente más aislados y cuentan con muchos menos recursos, lo que vuelve difícil la recolección de información demográfica.



Con este gráfico confirmamos que los outliers de los errores absolutos pertenecen a países con esperanza de vida baja. Adicionalmente, se presenta una pequeña concentración de casos atípicos en el rango de 60 a 70 y mayores a 80.

En términos generales, como se evidenció en la gráfica de distribución del error absoluto, la mayoría de los errores están entre 0 y 4 años.



Esta gráfica refuerza las observaciones realizadas previamente. Los outliers más grandes corresponden a predicciones que superaron en 7 o más años a valores en el rango de 40 a 55 años de esperanza de vida real. El segundo conjunto de predicciones con errores considerables está en el rango de 60 a 70 años, prediciendo valores

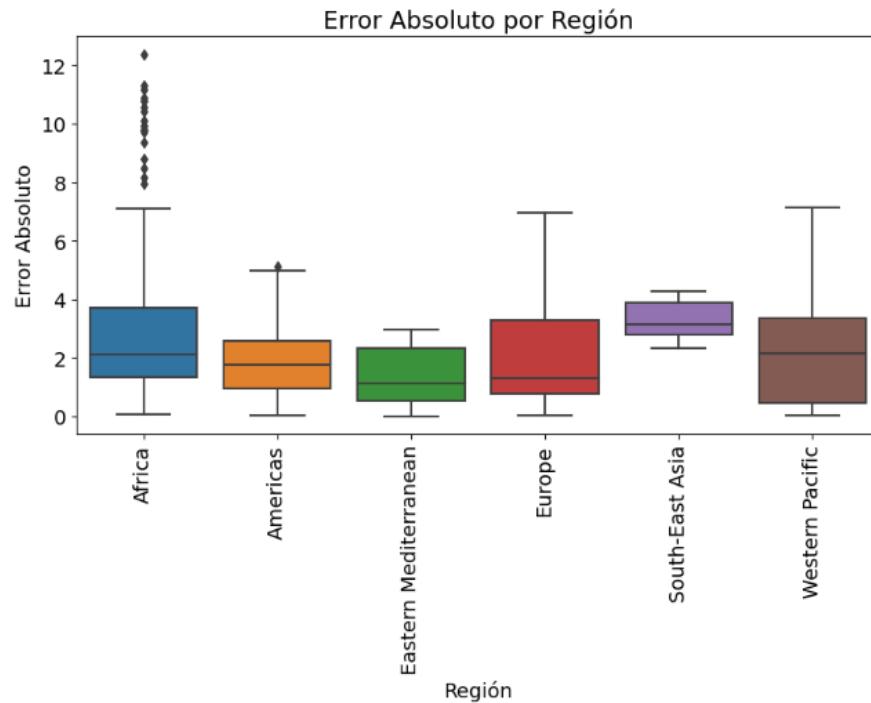
que superaban los reales por entre 5 y 8 años. Finalmente, el último grupo de outliers está en los límites superiores de esperanza de vida, también presentando errores de entre 5 y 8 años, pero esta vez por debajo del valor real.

Del comportamiento del modelo podemos identificar que las predicciones son en mayor proporción una subestimación de los valores reales. En otras palabras, son más las predicciones menores al valor real que las mayores a este. Asimismo, es claro que el modelo no tiene un buen rendimiento en los límites inferiores y superiores de esperanza de vida. Esto se debe a que los países con estos rangos de esperanza de vida son casos excepcionales alejados de la media.

Por un lado, cada vez son más escasos los países cuya esperanza de vida está por debajo de los 50 años. Aunque los registros no reflejen datos actuales, la tendencia del aumento sustancial en la esperanza de vida media se remonta a inicios del siglo XX. Mejoras en la higiene, el saneamiento, la nutrición, condiciones laborales, y acceso a recursos de supervivencia básicos, así como avances tecnológicos y en el campo de la medicina, fueron factores que presentaron un crecimiento exponencial en el siglo XX y que continúan en el XXI. Debido a esto, cada vez más países gozan de un aumento en la esperanza de vida de su población. En este orden de ideas, el modelo no tiene una alta precisión cuando se trata de países con esperanza de vida baja, pues fue entrenado con países que, en su mayoría, tienen una esperanza de vida superior a los 60 años.

En el caso opuesto, los países con esperanza de vida alta se están volviendo más comunes precisamente por la misma tendencia mencionada previamente. Sin embargo, todavía no son muchos los que cuentan con una esperanza de vida mayor a los 80 años. Por lo tanto, bajo la misma lógica de los países con baja esperanza de vida, el modelo fue entrenado con países que, en su mayoría, tienen una esperanza de vida menor a los 80 años.

Por último, se desconoce el causante del conjunto de outliers en el rango de 60 a 70 años. Estos pueden corresponder a ciertos países particulares cuyas esperanzas de vida están en este rango, pero sus características hacen que el modelo les asigne una esperanza de vida mayor a la real.

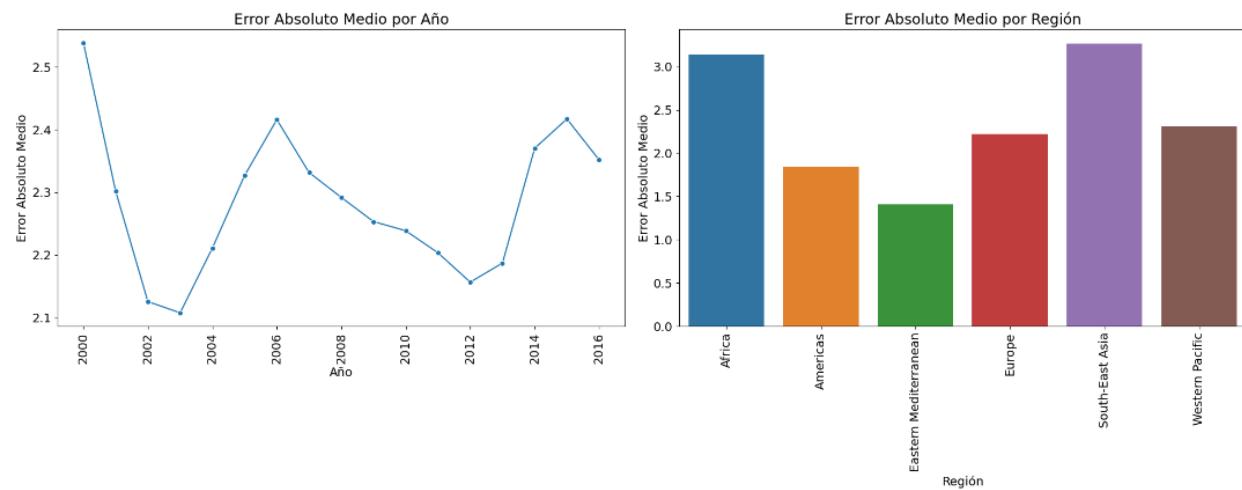


Estos boxplots confirman la segunda parte de la conjectura, la cual suponía que los outliers pertenecían mayoritariamente a países de África. Como se puede observar, el boxplot de la región de África presenta valores atípicos en la distribución de errores absolutos, alcanzando un valor máximo de poco más de 12 años.

Al analizar el conjunto de boxplots, se evidencia que, para todas las regiones, el 75% de los errores absolutos están por debajo de los 4 años. Para las regiones de las Américas, Mediterráneo Oriental y Europa, el 50% de los errores absolutos están por debajo de los 2 años, mientras que para África y el Pacífico Occidental, la mediana se encuentra levemente por encima de esta marca. En el caso del Sudeste Asiático, la mediana del error está por encima de los 3 años. Este es un caso particular, ya que a esta región solo pertenecen 11 países en el mundo, de los cuales uno fue eliminado del dataset por falta de datos. En este sentido, es lógico esperar un error mayor para los países del Sudeste Asiático, dado que el modelo no fue entrenado con suficientes datos de esta región.

Los amplios rangos intercuartílicos de los boxplots de África, Europa y el Pacífico Occidental revelan una alta variabilidad. Para las dos primeras regiones, esto puede deberse a su gran cantidad de países. Europa es la región con más países (53), seguida por África (47), lo que puede resultar en una gran diversidad reflejada en la variabilidad de la distribución. En el caso del Pacífico Occidental, aunque esta es la cuarta región con más países (27), precedida por las Américas (35), se asume que su alta variabilidad es consecuencia de la disparidad en la esperanza de vida de los países que la componen. Países como Australia, China, Nueva Zelanda y Japón tienen una esperanza de vida elevada que, en contraste con otros países como Camboya, Kiribati, Palau y Vietnam, es considerablemente superior, resultando en esta distribución. Siguiendo la lógica de que más países generan mayor variabilidad, las Américas también deberían evidenciar esta característica. Sin embargo, el estrecho rango intercuartílico de esta región indica que los países pertenecientes presentan una distribución más uniforme, lo que deriva en un mejor rendimiento del modelo.

Con todo esto en consideración, se puede afirmar que el modelo tiene un buen desempeño general para las regiones de las Américas, Mediterráneo Oriental y Europa; un desempeño intermedio para el Pacífico Occidental; y un desempeño relativamente bajo para África y el Sudeste Asiático.

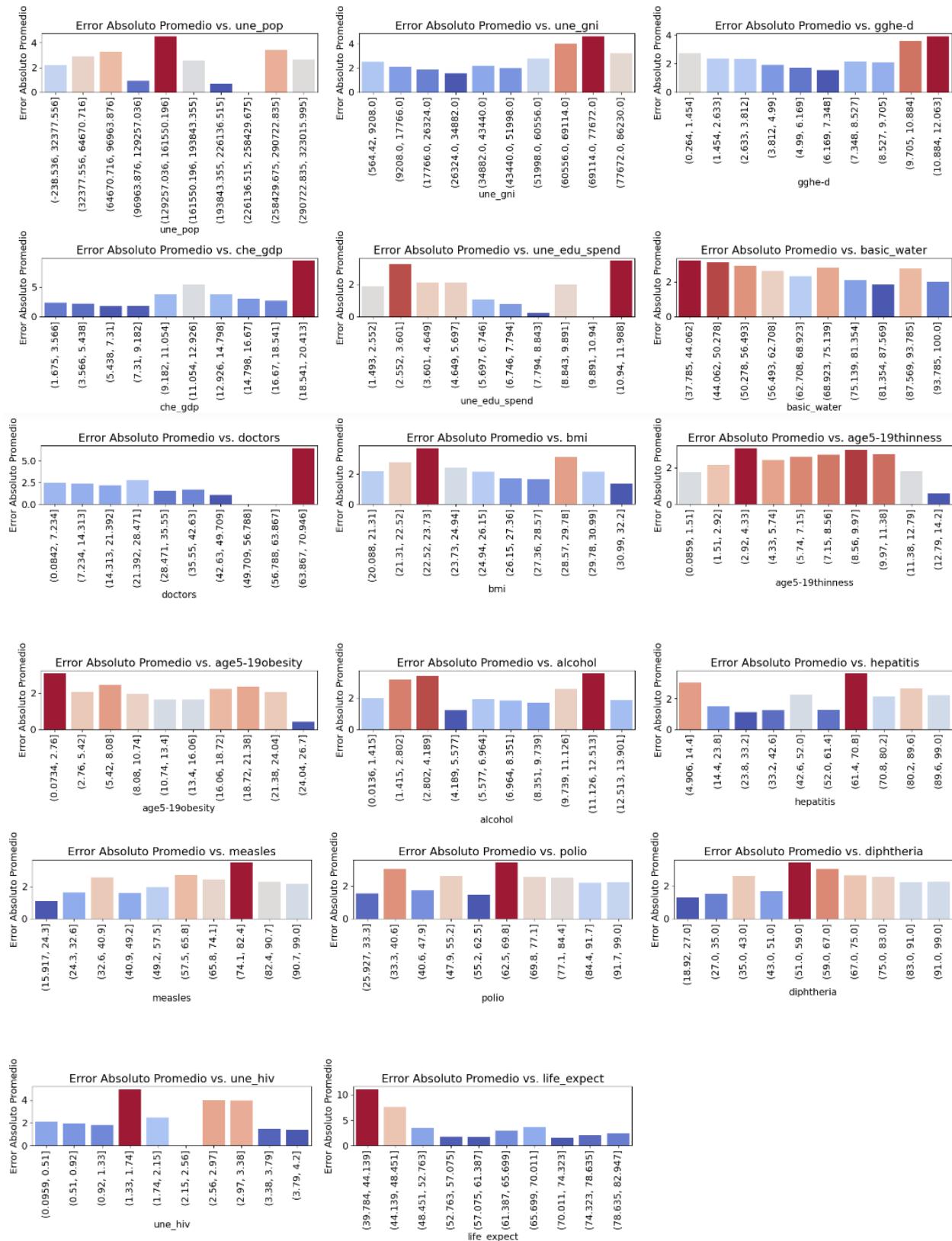


La primera gráfica advierte que no existe una relación entre los años y el error absoluto medio del modelo para los diferentes períodos. No hay una tendencia general creciente o decreciente que nos permita señalar para qué años es más preciso.

Aunque parezca que hay una alta variabilidad en el desempeño del modelo para los diferentes años, la mayor diferencia entre el mayor y menor error absoluto medio es de aproximadamente 0.45 años. En este orden de ideas, los errores absolutos medios son relativamente uniformes para todos los años.

La segunda gráfica confirma de manera general lo constatado en los boxplots anteriores. Las regiones de mayor error absoluto medio son el Sudeste de Asia y África, seguidos por el Pacífico Occidental, Europa, las Américas, y el Mediterráneo Oriental en ese orden.

Distribución del Error Absoluto Promedio por Variables Numéricas



De manera general, se puede apreciar que el modelo tiene un mal desempeño para los siguientes casos:

- Para valores bajos de las variables: basic_water, une_edu_spend, bmi, age5-19thinness, age5-19obesity, alcohol, y life_expect.
- Para valores intermedios de las variables: polio, diphtheria, y une_hiv.
- Para valores altos de las variables: une_gni, gghe-d, che_gdp, une_edu_spend, doctors, alcohol, hepatitis, y measles.

La variable une_pop parece no tener una tendencia marcada.

Cabe aclarar que estas apreciaciones son completamente subjetivas. Los juicios expuestos se basan en tomar los tres primeros rangos como valores bajos, los cuatro siguientes como intermedios, y los tres últimos como grandes, y realizar un análisis muy superficial de las tendencias. Estas afirmaciones están sujetas a duda, pues para conocer a detalle el comportamiento del modelo frente a cada variable, es necesario hacer un estudio más minucioso de los resultados y errores.

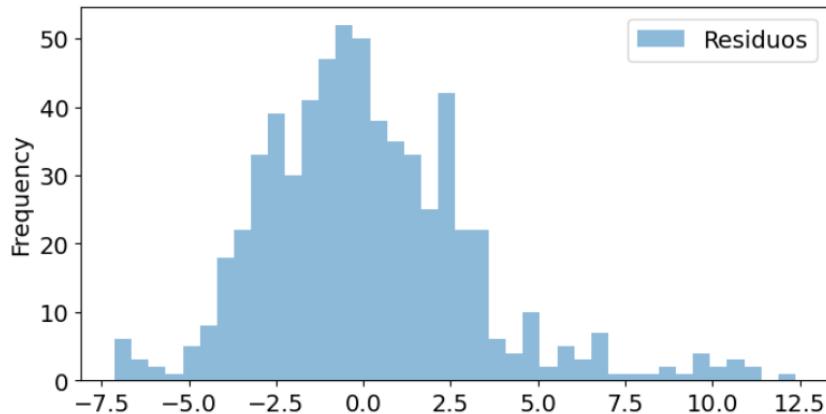
5.3 Intervalos de confianza para MAE, MSE y MAPE

Para hallar los intervalos de confianza de las métricas MAE, MSE y MAPE, se empleará un método no paramétrico llamado Bootstrap Percentile Confidence Intervals. Inicialmente, se calculan los residuos restandole a las predicciones los valores reales. Los resultados se convierten en valor absoluto, obteniendo así los errores absolutos (x). Al elevar estos errores absolutos al cuadrado (x^2), se obtienen los errores cuadráticos. Por último, los errores absolutos porcentuales ($x_percentage$) se obtienen dividiendo los errores absolutos por los valores reales.

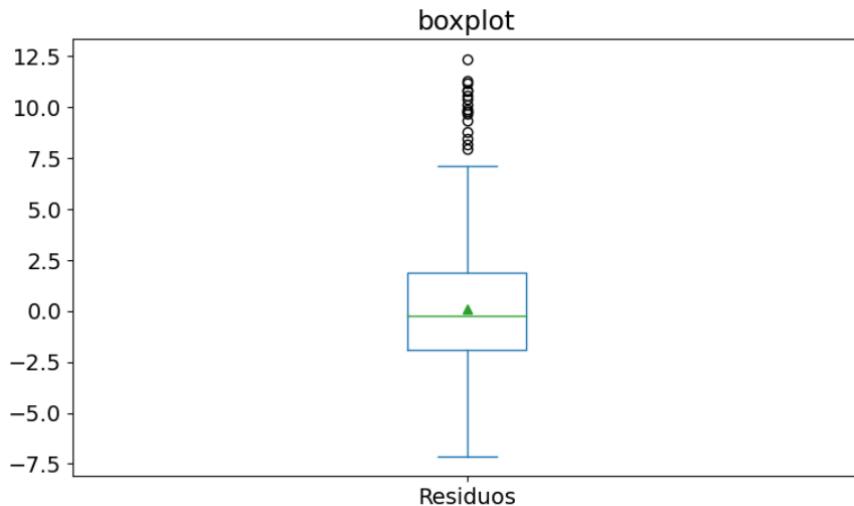
```
res = (pred_test_fm - y_test).values  
  
res_abs = np.abs(res)  
  
x = res_abs.copy()  
x2 = x**2  
x_percentage = x/abs(y_test)
```

La media de estas variables son equivalentes al MAE, MSE y MAPE como se observa a continuación:

```
print("MAE: ",x.mean())  
print("MSE: ",x2.mean())  
print("MAPE: ",x_percentage.mean())  
  
MAE: 2.2841096852078038  
MSE: 9.240226629727047  
MAPE: 0.034626150855223685  
  
print("MSE: " + str(mean_squared_error(y_test, pred_test_fm)))  
print("MAE: " + str(mean_absolute_error(y_test, pred_test_fm)))  
print("MAPE: " + str(mean_absolute_percentage_error(y_test, pred_test_fm)))  
  
MSE: 9.240226629727047  
MAE: 2.2841096852078038  
MAPE: 0.034626150855223685
```



Esta gráfica confirma que el modelo tiende a predecir más valores menores al valor real que mayores a este. Sin embargo, los errores positivos tienen una mayor cantidad de outliers que los negativos, además de un rango más amplio. Mientras que los errores negativos llegan a un máximo de aproximadamente -7, los positivos llegan a un poco menos de 12.5.



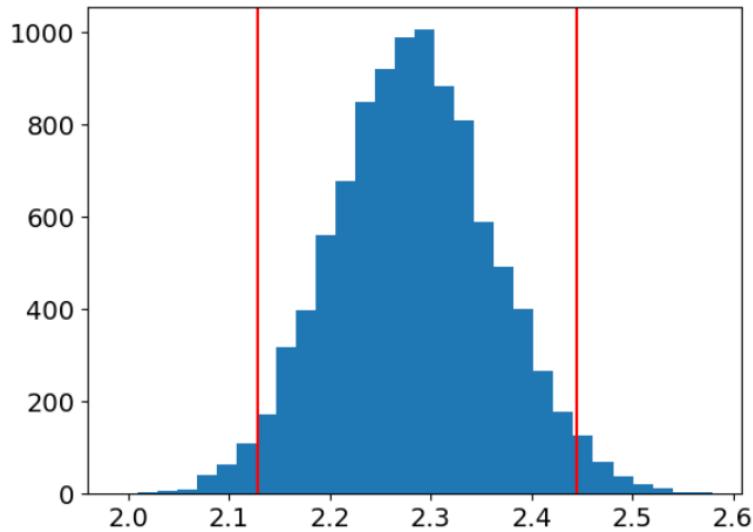
El boxplot confirma que el 50% de los errores están en el rango de -2.5 y 2.5.

Los intervalos de confianza Bootstrap son una técnica estadística que permite estimar la incertidumbre o variabilidad de una estadística calculada a partir de una muestra. Esta técnica se basa en el resampling (remuestreo) de la muestra original con reemplazo.

Para cada una de las métricas, se va a contar con una muestra de datos de tamaño $n = \text{len}(y)$, donde "y" puede ser x , $x2$, o $x_percentage$. Posteriormente, se generan 10000 nuevas muestras (bootstrap samples) de tamaño n a partir de la muestra original, seleccionando con reemplazo. A cada bootstrap sample se le calcula la estadística de interés, que, en este caso, es la media, y se almacena en una lista. Finalmente, se calculan los percentiles 2.5% y 97.5% de las medias de los bootstrap samples, que representan el límite inferior y superior del intervalo de confianza del 95% respectivamente.

5.3.1 Intervalo de confianza para MAE

```
Media muestra: 2.2841096852078038
Intervalo de confianza inferior: 2.1285539795881463
Intervalo de confianza superior: 2.4446389374901925
```

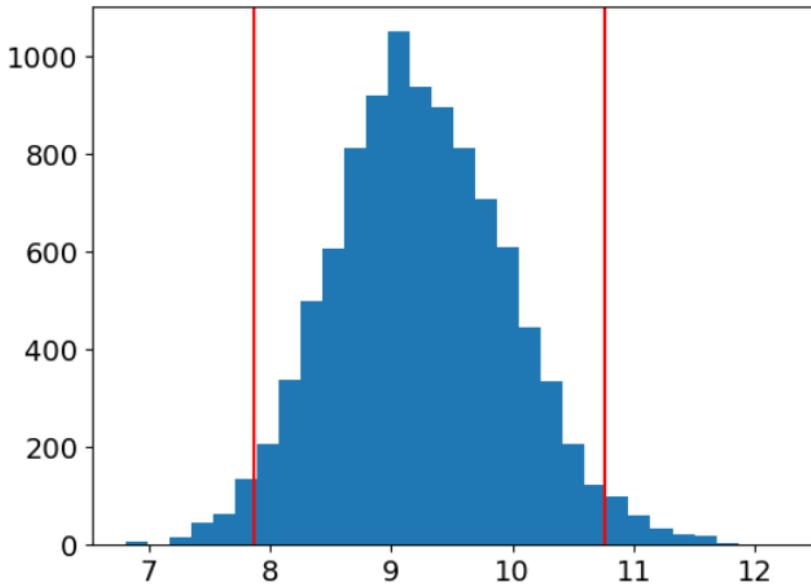


5.3.2 Intervalo de confianza para MSE

Media muestra: 9.240226629727047

Intervalo de confianza inferior: 7.8698692973518805

Intervalo de confianza superior: 10.756196204426418

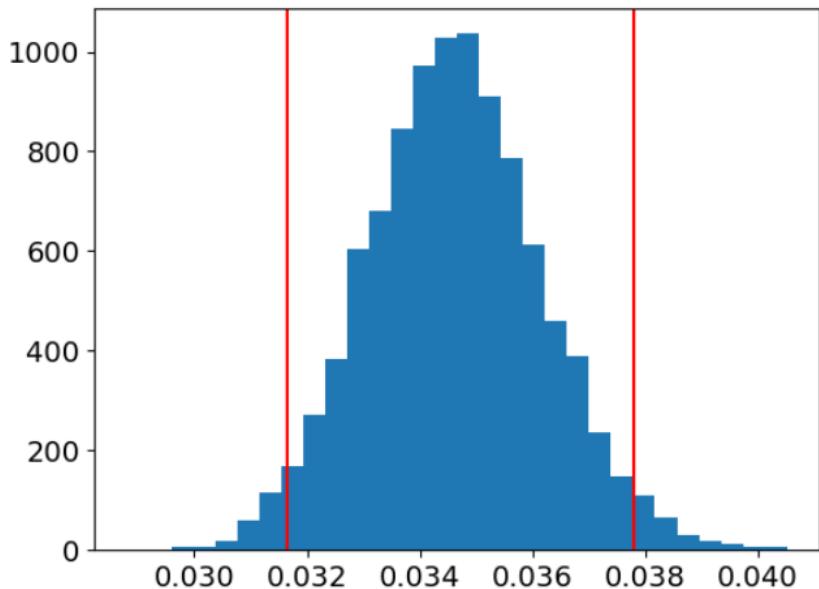


5.3.3 Intervalo de confianza para MAPE

Media muestra: 0.034626150855223685

Intervalo de confianza inferior: 0.03164420586545329

Intervalo de confianza superior: 0.03780043203549826



Con base en los resultados obtenidos, se determina que el intervalo de confianza no es demasiado amplio para las tres métricas, por tanto, se puede concluir que el modelo tiene una precisión y consistencia razonable y las predicciones no se desvían excesivamente de los valores reales.

5.4 A/B testing

5.4.1 Modelo final vs Modelo de prueba

Se aplicará el método de A/B testing, más específicamente, el test de Wilcoxon (Wilcoxon signed rank test), para comparar el modelo final con un modelo de prueba cuyas predicciones siempre serán la media simple de la variable de salida del conjunto de entrenamiento y validación.

El objetivo de este test es verificar que el modelo elegido si logra superar significativamente esta media simple, y con qué nivel de confianza se puede asegurar esto.

En el campo de la experimentación estadística, el A/B testing es usado para probar el efecto de algo, comparando un grupo de control contra un grupo de tratamiento. Para este caso puntual, se empleará para comparar el modelo final con el de prueba.

Las poblaciones a comparar serán los residuos absolutos del modelo final y el modelo de prueba. Ya que las dos poblaciones están emparejadas (cada modelo tiene los residuos para el mismo conjunto de datos), se utilizará una prueba de hipótesis no paramétrica para muestras emparejadas. El test de Wilcoxon plantea la hipótesis nula de que las dos poblaciones son equivalentes en términos de su mediana, y las alternativas, que las dos poblaciones difieren significativamente en su mediana (prueba de dos colas), o que una de ellas tiene una mediana significativamente mayor o menor que la otra (prueba de una cola).

```

train_val_mean = y_train_val.mean()
train_val_mean

69.11932963960231

err_abs_test_model = abs(train_val_mean - y_test)

stats.wilcoxon(x=err_abs_fm, y=err_abs_test_model, alternative='less')

WilcoxonResult(statistic=7624.0, pvalue=8.898483745399295e-90)

```

El p-valor de aproximadamente 0 indica que hay evidencia contundente para rechazar la hipótesis nula con un nivel de confianza superior al 99.9%. En otras palabras, los errores absolutos del modelo final en términos de su mediana son inferiores a los del modelo de prueba, por lo que se puede afirmar que el primero es significativamente mejor que el segundo.

5.4.2 Modelo final vs Segundo modelo de Gradient Boosting

Adicionalmente, se comprueba con qué nivel de confianza se seleccionó el modelo con la menor mediana de residuos absolutos entre el modelo final y el segundo modelo de Gradient Boosting.

```

gb_model = clone(best_gb_rs2)

gb_model.fit(x_train_val_selected, y_train_val)

pred_test_gb = gb_model.predict(x_test_selected)

err_abs_gb = abs(pred_test_gb - y_test)

stats.wilcoxon(x=err_abs_fm, y=err_abs_gb, alternative='less')

WilcoxonResult(statistic=84624.0, pvalue=0.0007677574483520307)

```

Los resultados indican con un nivel de confianza del 99.9% que la mediana de los errores absolutos del modelo final es menor que la del segundo modelo de Gradient Boosting. En este sentido, se confirma que, en términos de los errores absolutos, se seleccionó el mejor modelo entre los dos.

5.5 Evaluación del cumplimiento de los objetivos de negocio

El objetivo de negocio principal del proyecto es desarrollar un modelo predictivo que permita estimar la esperanza de vida de un determinado país a partir de características propias de este y de su población.

Los criterios de éxito son:

- Identificar los factores más significativos que afectan la esperanza de vida para proporcionar recomendaciones concretas a la OMS y a los gobiernos sobre dónde enfocar los esfuerzos para aumentar la longevidad y el bienestar de la población.
- Indicar posibles cambios en áreas de mejora que, al ser aplicadas, resulten en el aumento de la predicción de la esperanza de vida de un determinado país.

Con esto en mente, se puede concluir que el objetivo de negocio fue satisfecho. Además del desarrollo del modelo predictivo, se cumplieron a cabalidad los criterios de éxito. En las diferentes etapas del proyecto, se exploraron los diversos factores que afectan la esperanza de vida media al nacer de un país. Para cada uno de

estos factores, se profundizó en el tipo y magnitud del impacto que tienen sobre la predicción de la esperanza de vida, dando así respuesta a las **preguntas de negocio**. A partir de esta información, se pueden formular juicios bien fundamentados que sustenten proyectos gubernamentales respaldados por la OMS para aumentar la longevidad y el bienestar de las poblaciones.

Adicionalmente, la implementación de los valores Shapley en la etapa de despliegue permite identificar los posibles cambios en las diferentes variables que se pueden realizar para aumentar la predicción de la esperanza de vida de un país. Esto puede servir como base para iniciativas oficiales que, correctamente implementadas, resulten en un aumento real de la esperanza de vida de una población específica.

Por otra parte, los objetivos de minería de datos son:

1. Determinar patrones, correlaciones y tendencias entre las diferentes variables y la esperanza de vida.
2. Desarrollar un modelo predictivo que permita estimar la esperanza de vida de un país con base en factores clave relacionados con inmunizaciones, mortalidad, economía, sociedad y otros indicadores de salud, cuyo MAE sea el mínimo posible.

Y los respectivos criterios de éxito son:

1. Desarrollo de un modelo predictivo con el mínimo MAE sobre el conjunto de prueba entre distintos modelos entrenados.
2. Selección del mejor modelo a partir de los errores absolutos con mínimo 95% de confianza.

A partir de esto, con la misma justificación previamente expuesta, se concluye que el primer objetivo fue cumplido.

El segundo objetivo fue igualmente satisfecho, cumpliendo con ambos criterios de éxito. Entre los 27 modelos entrenados, el modelo final propuesto presenta el mínimo MAE, con un valor de 2.2841096852078038 sobre el conjunto de prueba. Dependiendo del proyecto que se vaya a desarrollar, este valor puede o no estar dentro del rango permisible.

Independiente de cuál sea el caso, se concluye que la precisión y usabilidad del modelo puede ser mejorada de las siguientes maneras:

1. Aumentando la cantidad de variables significativas para la predicción de la esperanza de vida.
2. Aumentando el total de registros al incluir los datos de los últimos 7 años.
3. Distribuyendo de manera proporcional las regiones en los conjuntos de entrenamiento, validación y prueba (eliminar distribución aleatoria) o hacer una distribución proporcional con base a las esperanzas de vida promedio de los países durante el periodo de 2000-2016.
4. Eliminando “Africa” como variable para impedir sesgo (habitar en África no es causal de menor Esperanza de vida).
5. Obtener los datos faltantes de los países por medios oficiales para reducir la proporción de datos imputados.
6. Entrenar otros modelos de ML como redes neuronales.
7. Utilizar la herramienta GridSearchCV para afinar los hiperparámetros del modelo y obtener el mejor rendimiento posible.

5.6 Exportación del modelo final

Para usar el modelo final en el despliegue, este es reentrenado con el conjunto de entrenamiento, validación y prueba combinados. Una vez entrenado, se guarda el modelo final y el escalador en archivos pickle.

```

saved_model = final_model.fit(data_x, data_y)

pk.dump(saved_model, open("final_model.pkl", 'wb'))

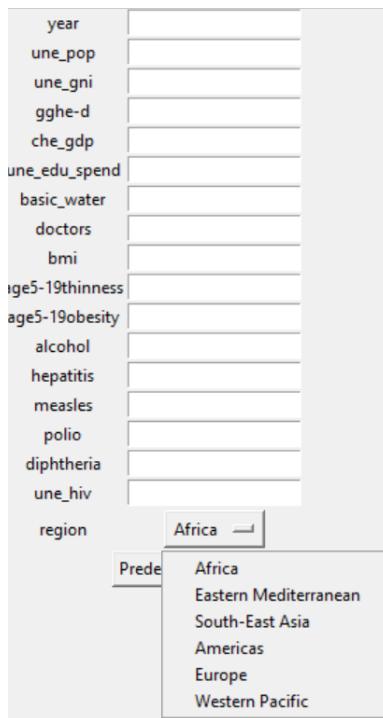
saved_scaler = StandardScaler().fit(df_numeric)

pk.dump(saved_scaler, open("scaler.pkl", 'wb'))

```

6. Despliegue

Para el despliegue se crea una interfaz gráfica, la cual, internamente realiza un preprocesamiento de los datos ingresados por el usuario para que puedan ser estandarizados y utilizados en la predicción. Finalmente, con base en la predicción, se muestra un gráfico de los valores SHAP que permite identificar el impacto que tuvo cada variable en la predicción.

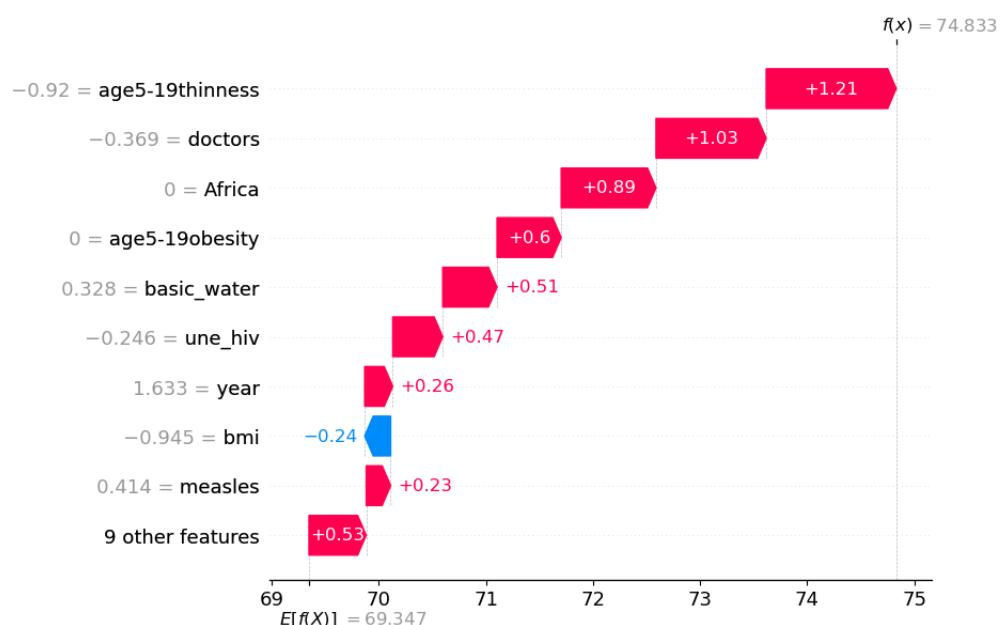


Predictor d...

year	2016
une_pop	50000
une_gni	10000
ghe-d	5
che_gdp	7
une_edu_spend	3
basic_water	90
doctors	10
bmi	23
age5-19thinness	1
age5-19obesity	6
alcohol	8
hepatitis	90
measles	92
polio	92
diphtheria	91
une_hiv	0.7
region	Europe

Predicción X

i La esperanza de vida predicha es: 74.83 años



Tecnología

En el marco de una implementación del proyecto en un ambiente productivo real, se plantea el siguiente desarrollo. En vista de la disponibilidad tecnológica y el presupuesto, las soluciones propuestas pueden estar sujetas a cambios, utilizando otras herramientas homólogas ofrecidas por otro proveedor. Por ejemplo, si las herramientas de AWS no resultan oportunas para el caso de uso, se pueden reemplazar con otras suministradas por Microsoft u otro proveedor relevante.

El proyecto es de baja complejidad y varias de sus etapas podrían realizarse de forma manual y local, como la ingesta, procesamiento, y entrenamiento del modelo. Sin embargo, en un caso donde se requiera una interfaz completamente automatizada y en la nube, se podría emplear AWS Step Functions para orquestar los servicios de AWS utilizados en el proyecto, y configurarlo con EventBridge para que ejecute todas las etapas una vez al año.

De igual forma, en el ciclo de vida de los datos se harán explícitas las soluciones dependiendo del caso de uso entre estos dos: desarrollo de alta complejidad en nube con alta automatización, integración y escalabilidad o desarrollo de baja complejidad parte local con intervención manual, baja integración y escalabilidad. No obstante, se hace la aclaración que, tomando como prioridad el costo-beneficio y la eficiencia, un desarrollo de baja complejidad con intervención manual y parte local, se adapta mucho mejor a las características y alcance del proyecto.

Ciclo de vida de los datos y procesamiento analítico

1. Ambiente tecnológico

Servicio de nube de AWS.

2. Origen de los datos

Los datos provienen de dos fuentes principales:

- **GHO** (Global Health Observatory), el cual es un recurso público habilitado por la Organización Mundial de la Salud (WHO por sus siglas en inglés), que ofrece datos y estadísticas sobre una amplia gama de indicadores de salud.
- **UIS** (UNESCO Institute for Statistics), el cual es el organismo de estadística oficial de la UNESCO y la principal fuente de datos relacionadas con educación, ciencia, cultura, y otros indicadores en los campos de acción de la UNESCO a nivel internacional.

De la combinación de la información ofrecida por ambos recursos, se construyó el conjunto de datos utilizado en este proyecto. En un escenario de implementación real, se tendría que acceder a estos datos por medio de los portales de cada institución.

3. Ingesta

Para la ingestión de datos, se postula una arquitectura tipo batch que incluya un proceso de ETL (Extract, Transform, Load). Esta arquitectura es relevante para el presente caso de uso, puesto que tiene en consideración el factor temporal del proyecto, que en este caso sería una ingestión anual (cada vez que las instituciones actualicen los indicadores para el año previo).

Para conectarse a las APIs públicas de estos portales y extraer la información de manera automática, se podría disponer del servicio AWS Lambda o con agentes ya sea de Elastic o de otros proveedores. Estos se encargarían de extraer los datos y almacenarlos en una zona transient en AWS S3. De igual forma, debido a la periodicidad,

reducida cantidad de datos, y baja complejidad del proyecto, la descarga podría hacerse de manera manual para un desarrollo de menor complejidad.

En la transformación, se combinarían los diferentes conjuntos de datos en uno solo que agrupe los indicadores para cada país y el año de interés, ya sea empleando un Job ETL de AWS Glue o un script local de Python que consuma los archivos almacenados en la zona transient del S3 y los cargue posteriormente a la zona raw. De nuevo, esto depende del nivel de robustez y automatización que se requiera.

Una vez combinados los conjuntos de datos para el año extraído, se realizaría la limpieza y preparación de los datos, para finalmente concatenarlos con el dataset maestro, almacenado en la zona trusted del S3, por medio de un Job de AWS Glue o un script de Python en local. Este maestro podría sobre escribirse o versionarse dependiendo de la necesidad.

4. Almacenamiento

Para un almacenamiento flexible, barato, escalable, seguro, y de alto rendimiento, se utilizaría el datalake AWS S3. Este estaría dividido en las siguientes zonas:

- **Zona transient:** Se almacenan los conjuntos de datos extraídos de las APIs de interés.
- **Zona raw:** Se almacenan los conjuntos de datos correspondientes a un año que combinan los conjuntos de datos de indicadores almacenados en la zona transient.
- **Zona trusted:** Se almacena el conjunto de datos maestro, el cual concatena todos los conjuntos de datos preparados y limpiados, correspondientes a los diferentes años de interés.
- **Zona refined:** Se almacenan los resultados del EDA (análisis exploratorio de datos por sus siglas en inglés) para ser consumidos en visualizaciones de BI. Igualmente, de ser necesario, almacena los resultados del modelo entrenado para su evaluación.

De ser necesario, para la catalogación automática de los datos, se emplearía AWS Glue, el cual ofrece un schema-on-read para los datos estructurados del proyecto. Estas tablas serían guardadas en una base de datos que posteriormente se leería en la etapa de procesamiento.

5. Procesamiento

Para el procesamiento, existen varias opciones. En el caso de alta complejidad y automatización, se podría aplicar la combinación de Amazon Redshift junto con Spectrum, la cual permite realizar análisis de los datos por medio de consultas SQL a gran escala. No obstante, este servicio estaría infrautilizado y representaría un gran costo para el presente caso de uso.

La siguiente opción, que se adaptaría más al caso de uso, sería disponer de Athena para realizar el mismo análisis exploratorio de los datos, ejecutando diferentes consultas SQL y almacenando los resultados en la zona refined.

Finalmente, para un desarrollo de baja complejidad, se pueden ejecutar manualmente scripts de Python que se conecten a la zona trusted, realicen todo el análisis exploratorio, y carguen los resultados a la zona refined del S3.

Dentro del procesamiento, se debe considerar el entrenamiento y reentrenamiento de los modelos. Considerando que el objetivo final es habilitar a los entes gubernamentales un modelo de aprendizaje automático que prediga la esperanza de vida de un país a partir de diversos indicadores, y facilite la toma de decisiones en inversiones y campañas nacionales y regionales, es fundamental que el modelo permanezca refinado a los cambios anuales a nivel mundial. Por ello, el entrenamiento de nuevos modelos de mayor rendimiento, y el reentrenamiento del modelo en producción con nuevos datos, debe hacer parte de la etapa de procesamiento.

Para un desarrollo de mayor robustez, que se integre a la suite de AWS, AWS SageMaker sería la opción ideal, admitiendo entrenamiento batch (on-demand o programado), scripts personalizados, despliegue a otros servicios,

y fácil integración con las demás herramientas de AWS. Si se quiere una solución de menor complejidad, ya que el caso de uso corresponde a un problema de pocos datos, el entrenamiento del modelo no requiere de muchos recursos ni de servicios optimizados para Big Data.

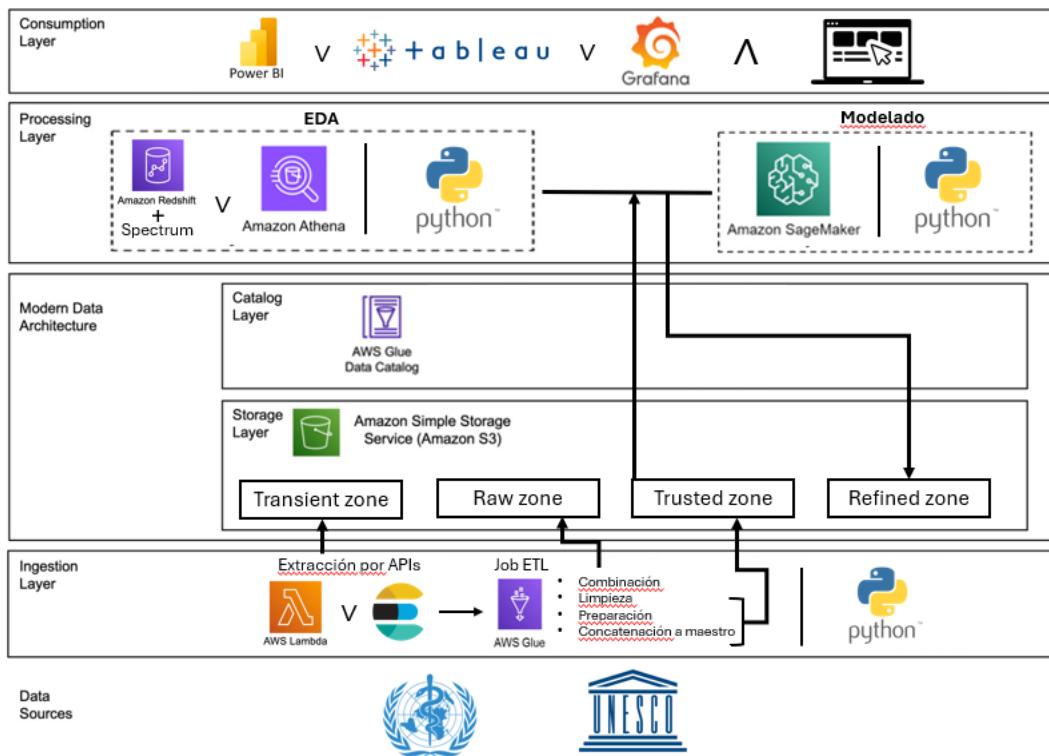
En ambos casos, tras la extracción anual de información, se alimentarían los nuevos datos al conjunto de entrenamiento del modelo, e igualmente, se evaluaría su rendimiento para comprobar que está siguiendo un desempeño acorde a los objetivos de negocio. De no ser así, se entrenarían otros modelos alineados con el estado del arte, para mantener un rendimiento competitivo y relevante. El modelo definitivo sería posteriormente almacenado en la zona refined del S3, junto a sus resultados en el conjunto de pruebas, para así explorarlos, tomar decisiones de negocio y generar visualizaciones. Finalmente, este sería desplegado en una REST API para su consumo desde la aplicación, que, en el caso de AWS, sería por medio de un endpoint en SageMaker.

6. Aplicaciones

Las visualizaciones del EDA y los resultados del modelo se consumirían en dashboards que se conectan a la zona refined del S3 con servicios como Power BI, Tableau, Grafana o, si se quiere una herramienta integrada directamente con AWS, Quicksight.

El modelo de aprendizaje automático se consumiría desde una aplicación simple en la web, donde el usuario ingresaría sus datos, la aplicación obtendría la API, enviaría los datos, recibiría la predicción como resultado, y se la presentaría al usuario acompañada de visualizaciones de valor, como los valores SHAP de la predicción.

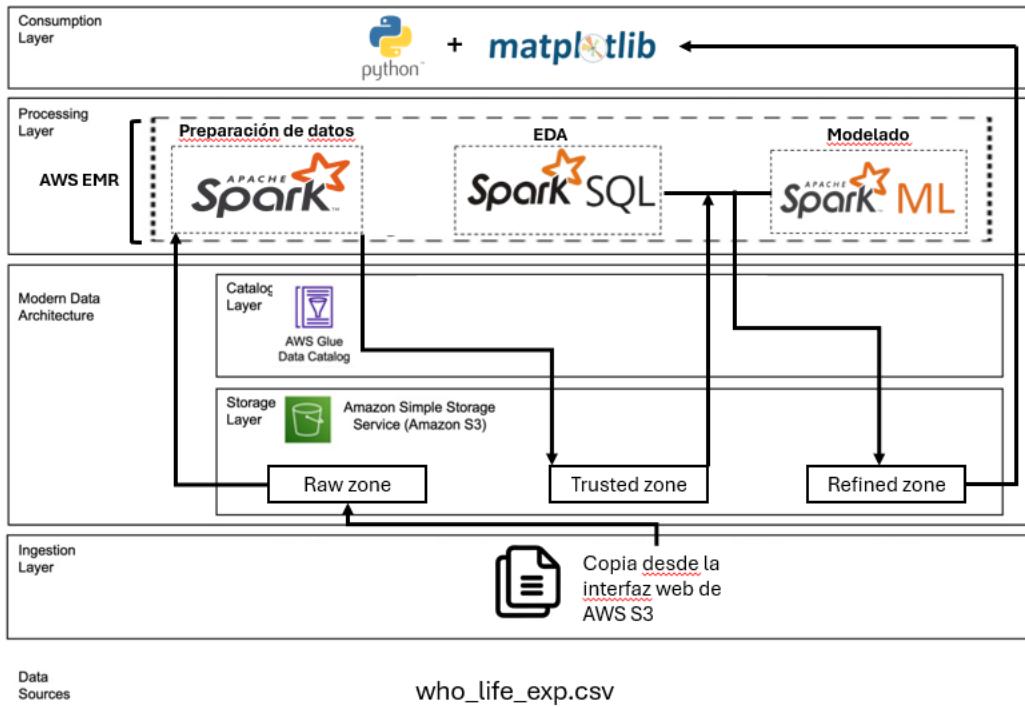
Gráfico



Despliegue del modelo en un caso hipotético de Big Data

Para replicar esta implementación, acceder al repositorio de GitHub: https://github.com/AAG0130/AAG0130-Proyecto-1-Parte-2_ST1800. Allí se encuentran todos los recursos necesarios para ello, incluyendo una guía con el paso a paso.

La metodología propuesta se resume en el siguiente gráfico:



Expectativas analíticas

La implementación del ciclo de vida expuesto en este despliegue se basa en el supuesto de que el caso de uso pertenece a la categoría de Big Data aunque realmente el conjunto de datos sea pequeño. En este sentido, las etapas de ingestión, almacenamiento, procesamiento y aplicaciones son desarrollados con herramientas de AWS y lenguajes de programación acordes, siguiendo una arquitectura tipo batch. Entre ellos, AWS S3 para el almacenamiento, EMR para el procesamiento, empleando Spark para la preparación de datos y AWS Glue para su catalogación, SparkSQL para el EDA, y, finalmente, SparkML para el entrenamiento del modelo.

Toda la etapa de procesamiento de este despliegue es efectuada en concordancia con el procesamiento realizado para este proyecto, pero adaptándolo al caso de uso de Big Data en cuestión. Por tanto, las librerías empleadas en el procesamiento del proyecto, como Pandas, Numpy y Scikit-learn, en un kernel de Python, son reemplazadas por SparkSQL y SparkML en un kernel de PySpark. En este orden de ideas, y considerando las diferencias en la aplicabilidad y las funciones habilitadas por ambos grupos de librerías, se esperan diferencias en dos subprocessos del ciclo de vida: primero, en la imputación de los datos faltantes, y segundo, en el entrenamiento del modelo de aprendizaje automático.

La primera se debe a la diferencia en los métodos de imputación ofrecidos por pandas.interpolate y pyspark.sql.window. Mientras el primero realiza interpolación numérica (lineal, polinomial, etc.) estimando los valores faltantes (NaN) en función de valores adyacentes (Geeks for Geeks, 2024), el segundo imputa datos por grupo o partición, aplicando funciones de agregación o relleno como el promedio o la suma (Geeks for Geeks, 2022).

La segunda diferencia parte de la disimilitud entre los parámetros aceptados por sklearn.ensemble.RandomForestRegressor y pyspark.ml.regression.RandomForestRegressor. Para la selección del modelo óptimo en el proyecto, se entrenaron 9 modelos de regresión, cada uno sobre 3 conjuntos de datos distintos: todas las variables, variables seleccionadas, y primeros 6 componentes principales. Cada uno de ellos se entrenó utilizando el método de RandomizedSearchCV, y se llegó a la conclusión de que el modelo con mejor

desempeño fue el Random Forest Regressor para el conjunto de variables seleccionadas con Backward Selection, y con una serie de hiperparámetros definidos por el RandomizedSearchCV. Este mismo caso será implementado en la presente implementación con SparkML, sin embargo, por las dos diferencias previamente mencionadas, se anticipan resultados distintos en el desempeño del modelo.

Por otra parte, en este mismo subprocesso, se espera una variación en el tiempo de entrenamiento del modelo con SparkML versus Scikit-learn. Considerando el tamaño del conjunto de datos usado, el cual consta tan solo de 3111 registros (sin todavía realizar la limpieza de datos), Scikit-learn está mucho mejor adaptado a esta cantidad pequeña en comparación a SparkML, el cual está diseñado para procesamiento distribuido en clusters de Big Data cuando los datos no caben en memoria. En este orden de ideas, es probable que SparkML tarde más segundos en finalizar el entrenamiento que Scikit-learn.

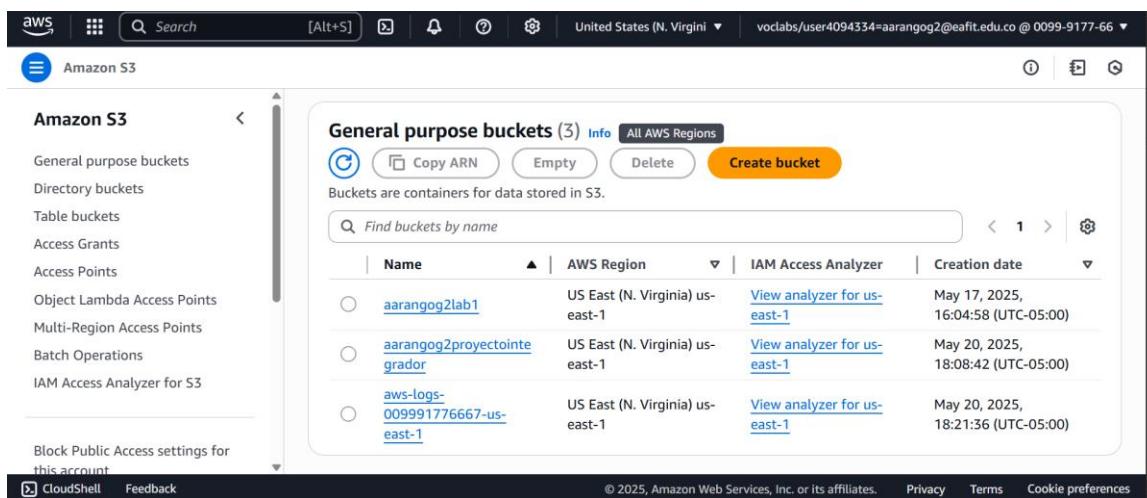
Aplicación del ciclo de vida

1. Fuentes de datos

La fuente de datos empleada será el archivo “who_life_exp.csv”.

2. Ingesta

Se crea un bucket S3 en AWS con el nombre “aarangog2proyectointegrador” para almacenar los archivos generados a lo largo del proyecto. Allí se hace la carga manual del archivo “who_life_exp.csv” por medio de la interfaz web de AWS S3.



The screenshot shows the AWS S3 console interface. On the left, there is a sidebar with navigation links: General purpose buckets, Directory buckets, Table buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, and IAM Access Analyzer for S3. Below this is a section for 'Block Public Access settings for this account'. The main area is titled 'General purpose buckets (3)' and contains a table with three rows. The table has columns for Name, AWS Region, IAM Access Analyzer, and Creation date. The first two rows correspond to the buckets mentioned in the text, and the third row is for log files. The IAM Access Analyzer column contains links to view analyzers for each bucket.

Name	AWS Region	IAM Access Analyzer	Creation date
aarangog2lab1	US East (N. Virginia) us-east-1	View analyzer for us-east-1	May 17, 2025, 16:04:58 (UTC-05:00)
aarangog2proyectointegrador	US East (N. Virginia) us-east-1	View analyzer for us-east-1	May 20, 2025, 18:08:42 (UTC-05:00)
aws-logs-00991776667-us-east-1	US East (N. Virginia) us-east-1	View analyzer for us-east-1	May 20, 2025, 18:21:36 (UTC-05:00)

3. Almacenamiento

Dentro de este S3 se crean tres zonas: zona raw, zona trusted y zona refined. En la primera (zona raw) se almacena la fuente de datos original: “who_life_exp.csv”. En la segunda (zona trusted), los datos preparados, producto del proceso de preparación de datos realizado en el EMR. Estos serán empleados para el EDA y entrenamiento del modelo, guardando sus resultados en la última zona (zona refined), junto con el modelo y el escalador utilizado en la estandarización de los datos.

Bucket 1: aarangog2proyectointegrador

Name	Type	Last modified	Size	Storage class
install-my-jupyter-libraries.sh	sh	May 22, 2025, 12:40:44 (UTC-05:00)	92.0 B	Standard
jupyter/	Folder	-	-	-
zona raw/	Folder	-	-	-
zona refined/	Folder	-	-	-
zona trusted/	Folder	-	-	-

Bucket 2: aarangog2proyectointegrador/zona raw/

Name	Type	Last modified	Size	Storage class
who_life_exp.csv	csv	May 20, 2025, 18:10:10 (UTC-05:00)	683.7 KB	Standard

4. Procesamiento

Para el procesamiento, se crea un cluster con el servicio AWS EMR, definiendo una configuración que habilita el uso de distintos servicios. Dentro de estos, se emplea Jupyter Hub para crear los notebooks requeridos para la preparación de datos, análisis exploratorio de datos (EDA), y entrenamiento del modelo de aprendizaje automático, llamados “Data prep”, “EDA” y “Train Model” respectivamente. En el primero se realiza la organización, limpieza y transformación de los datos por medio de Apache Spark; en el segundo se utiliza en mayor medida la librería de SparkSQL para realizar consultas sobre las tablas, resultado de la preparación, que fueron catalogadas con AWS Glue; y finalmente, en el tercero, se construye un modelo de regresión de Random Forest con SparkML.

The screenshot shows two main sections. The top section is the Amazon EMR console under the 'EMR on EC2' category, displaying a list of 10 clusters. The bottom section is the JupyterHub interface, showing a list of running notebooks.

Amazon EMR - Clusters (10)

Cluster ID	Cluster name	Status	Creation time (UTC-05:00)	Elapsed time
j-20HUFUVXBNC	aarangog2 Proyecto Integrador	Waiting	May 24, 2025, 20:07	45 minutes, 43 seconds
j-199510C06VKXQ	aarangog2 Proyecto Integrador	Terminated	May 24, 2025, 18:46	46 minutes, 53 seconds
j-1POY9B9V9RNXGY	aarangog2 Proyecto Integrador	Terminated with errors	May 24, 2025, 18:36	5 minutes, 56 seconds
j-3HD0VS2N0H08	aarangog2 Proyecto Integrador	Terminated with errors	May 24, 2025, 16:07	2 hours, 25 minutes
j-2K46QHL799PV8	aarangog2 Proyecto Integrador	Terminated	May 24, 2025, 14:54	1 hour, 17 minutes
j-1KBRMCMUTAB5KG	aarangog2 Proyecto Integrador	Terminated	May 24, 2025, 08:07	2 hours, 26 minutes
j-2IYI1UU771U2	aarangog2 Proyecto Integrador	Terminated	May 22, 2025, 19:12	1 hour, 9 minutes
j-1BKPF9EPP6DK79	aarangog2 Proyecto Integrador	Terminated	May 22, 2025, 12:59	39 minutes, 51 seconds

jupyterhub

File	Running	Clusters	Upload	New	
Select items to perform actions on them.			Name	Last Modified	File size
0			Data prep.ipynb	Running	21 minutes ago
			EDA.ipynb		4 hours ago
			Train Model.ipynb		an hour ago

4.1 Preparación de datos

En este subproceso de la etapa de procesamiento se generan varios resultados que son almacenados en la zona trusted del S3. Cada uno de estos resultados incluye tanto los pasos intermedios en la generación del conjunto de datos preparado para el entrenamiento del modelo, como el propio conjunto de datos final.

A continuación, se hace un listado de las transformaciones que sufre el conjunto de datos original a medida que pasa por cada paso intermedio. El proceso es incremental, lo que implica que para la generación de “data_filtered”, el insumo son los datos crudos “who_life_exp.csv”, para “data_selected” el insumo es “data_filtered”, y así sucesivamente hasta llegar a “data_prepared selected”.

Es importante recordar que se está replicando el proceso seguido en el proyecto, lo que implica que cada transformación está justificada por análisis exhaustivos documentados previamente en este documento.

4.1.1 Data_filtered

1. Reordenar las columnas para que tengan una organización más fácil de interpretar, y la variable objetivo “life_expect” en la última columna.
2. Aproximar a 100 todos los valores de la columna “basic_water” superiores a 100. “basic_water” representa el porcentaje de la población usando por lo menos servicios básicos de agua potable, por tanto, no puede tener valores superiores a 100.
3. Guardar el resultado en formato parquet en la carpeta “data_filtered” del S3 “aarangog2proyectointegrador”.

4.1.2 Data_selected

1. Eliminar las columnas: "une_life", "une_infant", "gni_capita", "hospitals", "une_literacy", "une_school", "une_poverty", "life_exp60", "infant_mort", "adult_mortality", y "age1-4mort", por diferentes motivos relacionados con la calidad de los datos.
2. Eliminar las filas correspondientes a los países ("country"): South Sudan", "Somalia", "Democratic People's Republic of Korea", "Sudan", por motivos de completitud de datos.
3. Guardar el resultado en formato parquet en la carpeta "data_selected" del S3 "aarangog2proyectointegrador".

4.1.3 Data_imputed

1. Imputar valores faltantes con la función avg() del método pyspark.sql.window, particionando las columnas faltantes según la variable "country". Por tanto, se llenaban los datos faltantes con el promedio de esta columna para ese país.
2. Imputar los valores faltantes que no pudieron ser imputados con el método anterior debido a la ausencia total de datos para esa columna, con el promedio de esa columna para la región ("region") a la que pertenece ese país.
3. Guardar el resultado en formato parquet en la carpeta "data_imputed" del S3 "aarangog2proyectointegrador".

4.1.4 Data_numeric

1. Eliminar las columnas categóricas: "country", "country_code", y "region".
2. Guardar el resultado en formato parquet en la carpeta "data_imputed" del S3 "aarangog2proyectointegrador".

4.1.5 Data_standard

1. Estandarizar los valores numéricos, exceptuando la variable objetivo "life_expect", con StandardScaler.
2. Agregar las columnas categóricas, asociando los valores a sus correspondientes filtas.
3. Guardar el resultado en formato parquet en la carpeta "data_standard" del S3 "aarangog2proyectointegrador".

4.1.6 Data_prepared

1. Aplicar one-hot encoding a la variable categórica "region".
2. Eliminar las columnas categóricas "region", "country", y "country_code".
3. Guardar el resultado en formato parquet en la carpeta "data_prepared" del S3 "aarangog2proyectointegrador".

4.1.7 Data_prepared_selected

1. Eliminar las columnas "Eastern Mediterranean", "South-East Asia", "Americas", "Europe", y "Western Pacific", resultado del one-hot encoding, tras realizar el proceso de ingeniería de características: Backward Selection.
2. Guardar el resultado en formato parquet en la carpeta "data_prepared_selected" del S3 "aarangog2proyectointegrador".

4.2 Catalogación con AWS Glue

Para ejecutar consultas sobre los resultados de la preparación de datos, se deben catalogar con AWS Glue. En otras palabras, debido a que los resultados fueron almacenados en un datalake, el cual tiene un esquema de lectura (schema-on-read), se deben crear los esquemas para cada uno de los archivos para realizar las consultas con SQL.

Inicialmente, se crea y corre un crawler por cada archivo resultado de la preparación de datos.

Name	State	Last run	Last run timestamp	Log	Table changes from l...
catalogdatafiltered	Ready	Succeeded	May 24, 2025 at 19:5...	View log	1 created
catalogdataimputed	Ready	Succeeded	May 24, 2025 at 19:5...	View log	1 created
catalogdatanumeric	Ready	Succeeded	May 24, 2025 at 19:5...	View log	1 created
catalogdataprepared	Ready	Succeeded	May 24, 2025 at 23:1...	View log	1 created
catalogdataprepares...	Ready	Succeeded	May 24, 2025 at 23:1...	View log	1 created
catalogdataselected	Ready	Succeeded	May 24, 2025 at 19:5...	View log	1 created
catalogdatastandard	Ready	Succeeded	May 24, 2025 at 23:1...	View log	1 created
catalogonu	Ready	Succeeded	May 17, 2025 at 21:3...	View log	2 created
catalogтик	Ready	Succeeded	May 17, 2025 at 21:4...	View log	7 created

Al correr los crawlers, se generan los esquemas, y las tablas, listas para ser consultadas, son guardadas en la base de datos “proyecto1db”.

Databases (4)

Name	Description	Location URI	Created on (UTC)
default	default database	hdfs://ip-172-31-79-174.ec2.internal:8020/user/spark/wai	May 24, 2025 at 21:42:42
labsdb	-	-	May 17, 2025 at 21:31:0C
myspectrum_db	-	-	May 17, 2025 at 23:13:34
proyecto1db	-	-	May 24, 2025 at 12:14:21

Database properties

Name	Description	Location	Created on (UTC)
proyecto1db	-	-	May 24, 2025 at 12:14:21

Tables (7)

Name	Database	Location	Classification	Deprecated	View data	Data quality	Column statistics
data_filtered	proyecto1db	s3://aarangog2/proyecto	Parquet	-	Table data	View data quality	View statistics
data_imputed	proyecto1db	s3://aarangog2/proyecto	Parquet	-	Table data	View data quality	View statistics
data_numeric	proyecto1db	s3://aarangog2/proyecto	Parquet	-	Table data	View data quality	View statistics
data_prepared	proyecto1db	s3://aarangog2/proyecto	Parquet	-	Table data	View data quality	View statistics
data_prepared_selected	proyecto1db	s3://aarangog2/proyecto	Parquet	-	Table data	View data quality	View statistics
data_selected	proyecto1db	s3://aarangog2/proyecto	Parquet	-	Table data	View data quality	View statistics
data_standard	proyecto1db	s3://aarangog2/proyecto	Parquet	-	Table data	View data quality	View statistics

4.3 EDA

El análisis exploratorio de datos se realizará únicamente sobre los conjuntos “data_filtered” y “data_selected”. El primero, porque contiene la totalidad de los países y variables originales, y solo se harán consultas en las variables que no tienen valores faltantes. El segundo, porque ya se eliminó los países con exceso de datos faltantes, lo que permite hacer consultas puntuales sobre el año más reciente para otras variables importantes.

Para ello, se leen las tablas directamente desde la base de datos “proyecto1db” con SparkSQL y se crean vistas temporales de estas mismas tablas para poder realizar las consultas necesarias sin modificar la estructura y contenido original.

Sobre “data_filtered” se hacen las siguientes consultas con SparkSQL:

- Promedio de "life_expect" por región y año:** Promedio de la variable "life_expect" para todos los países agrupados por la variable "region" y variable "year".
- Promedio de "adult_mortality" por región y año:** Promedio de la variable "adult_mortality" para todos los países agrupados por la variable "region" y variable "year".
- Promedio de "infant_mort" por región y año:** Promedio de la variable "infant_mort" para todos los países agrupados por la variable "region" y variable "year".

- **Promedio de "life_exp60" por región y año:** Promedio de la variable "life_exp60" para todos los países agrupados por la variable "region" y variable "year".
- **Promedio de "age1-4mort" por región y año:** Promedio de la variable "age1-4mort" para todos los países agrupados por la variable "region" y variable "year".
- **Cantidad de países únicos en data_filtered:** Cuenta del número de valores únicos de la columna "country".

Sobre "data_selected" se hacen las siguientes consultas con SparkSQL:

- **Cantidad de países únicos en data_selected:** Cuenta del número de valores únicos de la columna "country".
- **Países con "age5-19thinness" igual o mayor al 20% en el año 2016:** Países ("country") que para el año ("year") 2016 tienen un valor de la variable "age5-19thinness" igual o mayor a 20.
- **Países con "age5-19obesity" igual o mayor al 20% en el año 2016:** Países ("country") que para el año ("year") 2016 tienen un valor de la variable "age5-19obesity" igual o mayor a 20.
- **Países con "hepatitis" igual o menor al 80% en el año 2016:** Países ("country") que para el año ("year") 2016 tienen un valor de la variable "hepatitis" igual o menor a 80.
- **Países con "measles" igual o menor al 80% en el año 2016:** Países ("country") que para el año ("year") 2016 tienen un valor de la variable "measles" igual o menor a 80.
- **Países con "polio" igual o menor al 80% en el año 2016:** Países ("country") que para el año ("year") 2016 tienen un valor de la variable "polio" igual o menor a 80.
- **Países con "diphtheria" igual o menor al 80% en el año 2016:** Países ("country") que para el año ("year") 2016 tienen un valor de la variable "diphtheria" igual o menor a 80.
- **Países con "une_hiv" igual o mayor al 20% en el año 2016:** Países ("country") que para el año ("year") 2016 tienen un valor de la variable "une_hiv" igual o mayor a 20.

Cada uno de los resultados de estas consultas son guardados en la zona refined del S3 "aarangog2proyectointegrador".

4.4 Diseño e implementación del modelo de aprendizaje automático

La división de los conjuntos de entrenamiento y prueba se realiza con la proporción 80/20, generando el primer conjunto con la selección aleatoria del 80% del total de los países (considerando todos los datos entre sus períodos de 2000 a 2016), y asignando el otro 20% para el conjunto de prueba. Una vez generados, se entrena el modelo con los siguientes hiperparámetros:

```
rf = RandomForestRegressor(  
    featuresCol="features",  
    labelCol=target_col,  
    numTrees=144,  
    maxDepth=20,  
    minInstancesPerNode=5,  
    minInfoGain=0.0,  
    maxBins=32,  
    seed=semilla,  
    featureSubsetStrategy="log2",  
    bootstrap=False  
)
```

Estos fueron seleccionados buscando reflejar los mismos hiperparámetros utilizados en el modelo construido con Scikit-learn que se muestran a continuación, pero considerando las particularidades de SparkML:

```
rf_rs1.best_params_  
  
{'n_estimators': 144,  
 'min_samples_split': 5,  
 'min_samples_leaf': 1,  
 'max_features': 'log2',  
 'max_depth': 60,  
 'criterion': 'squared_error',  
 'bootstrap': False}
```

5. Aplicación

Para la aplicación, se diseña un notebook que se conecta directamente a la zona refined del AWS S3 para la visualización de los resultados del EDA y del desempeño del modelo. En este esquema de visualización, aunque no está diseñado para ser un producto final de cara al cliente, es una aproximación de los diversos gráficos que tendría a su disposición.

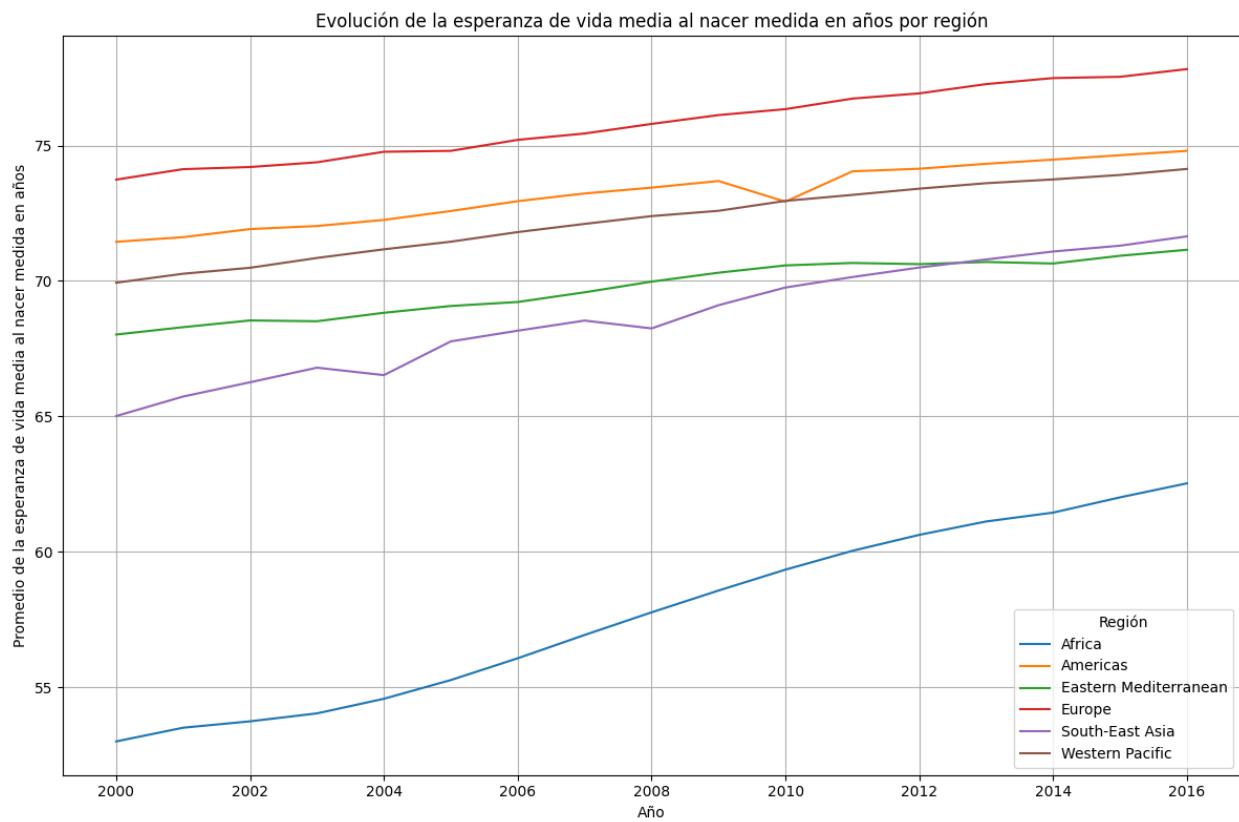
En el notebook se deben ingresar las siguientes credenciales de la sesión de AWS:

- aws_access_key_id
- aws_secret_access_key
- aws_session_token

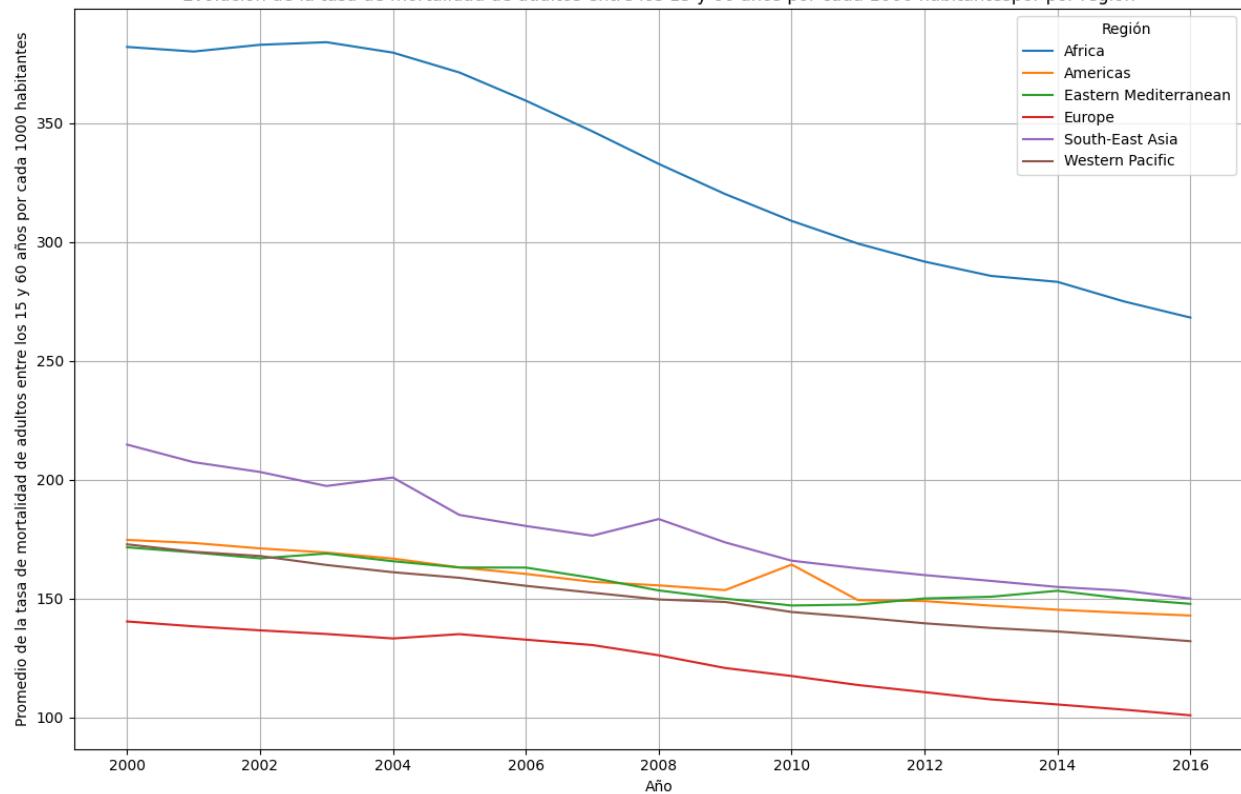
Así como el path hacia la zona refined del S3, que, en nuestro caso, sería: "s3://aarangog2proyectointegrador/zona refined/".

Asumiendo que el usuario guardó los archivos con los mismos nombres definidos en las fases de EDA y diseño e implementación del modelo de aprendizaje automático, solo se debe correr el notebook y se obtendrían las siguientes visualizaciones:

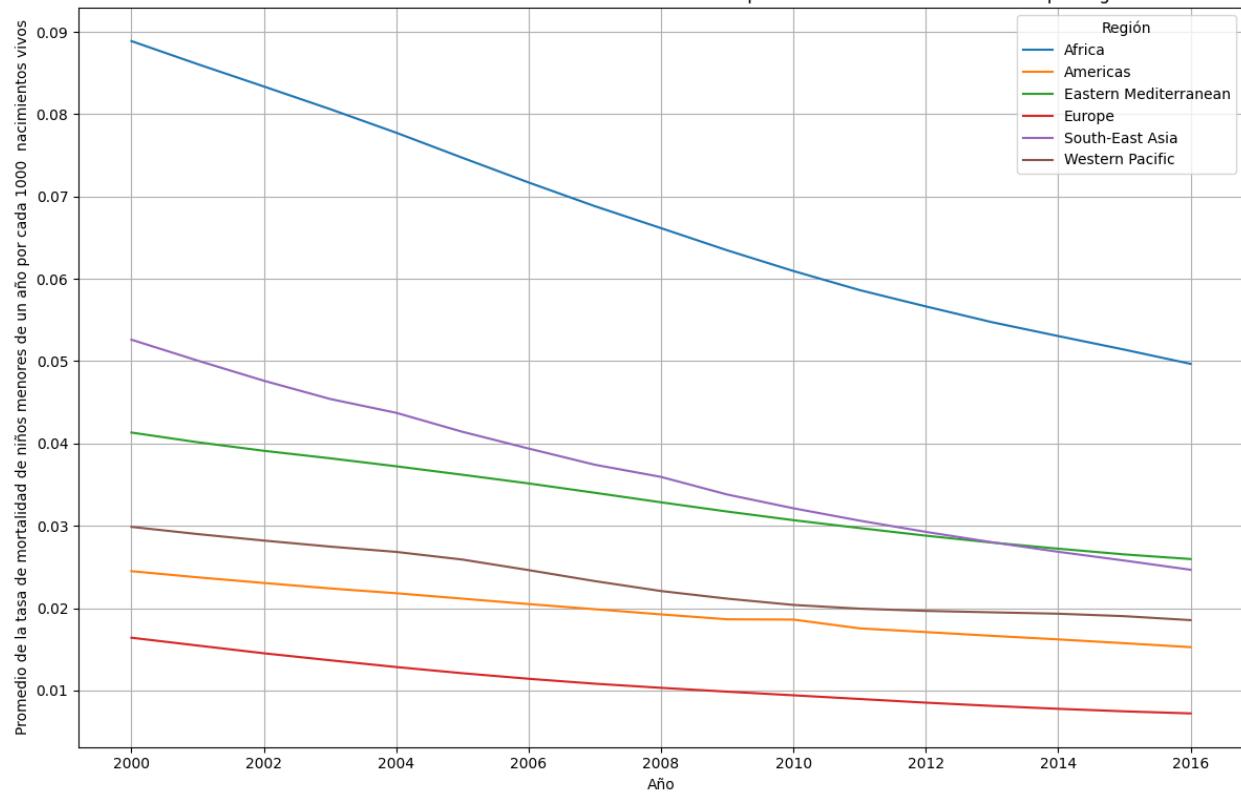
Para “*data_filtered*”:

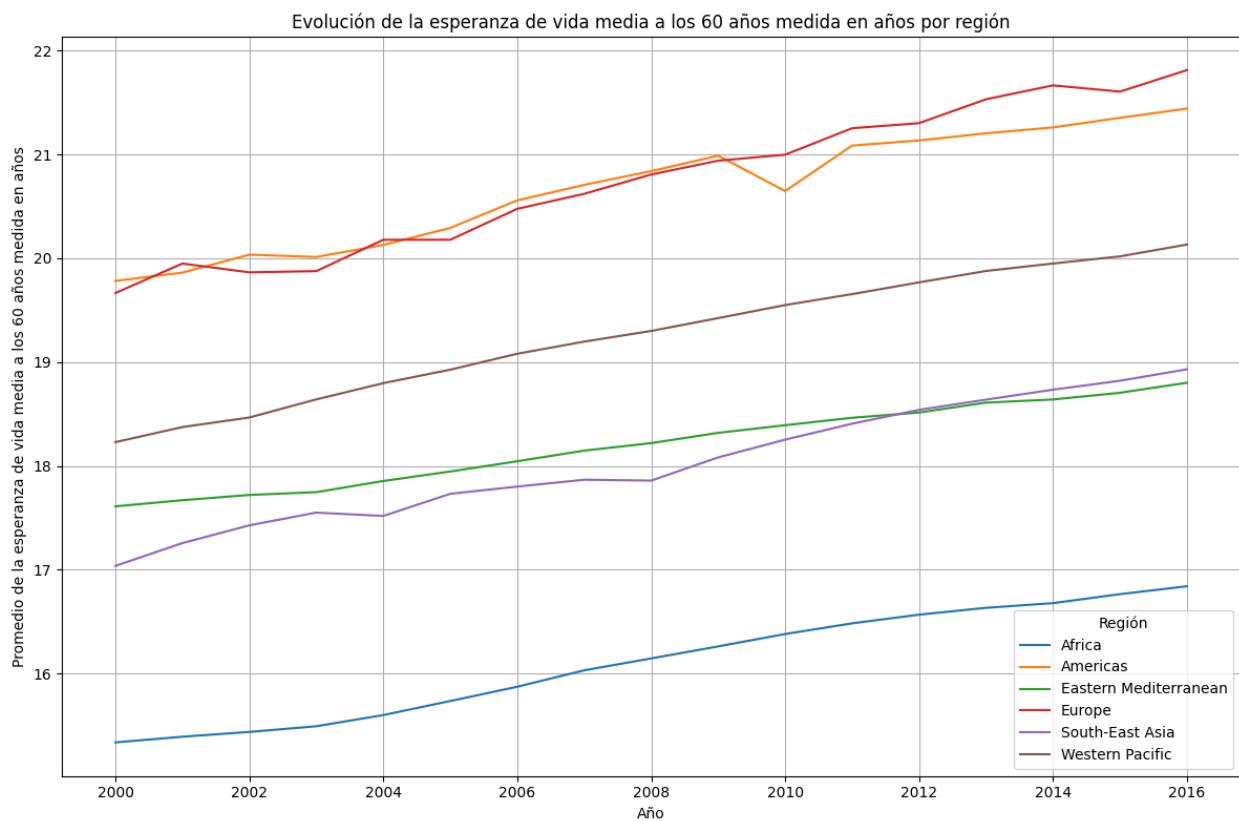
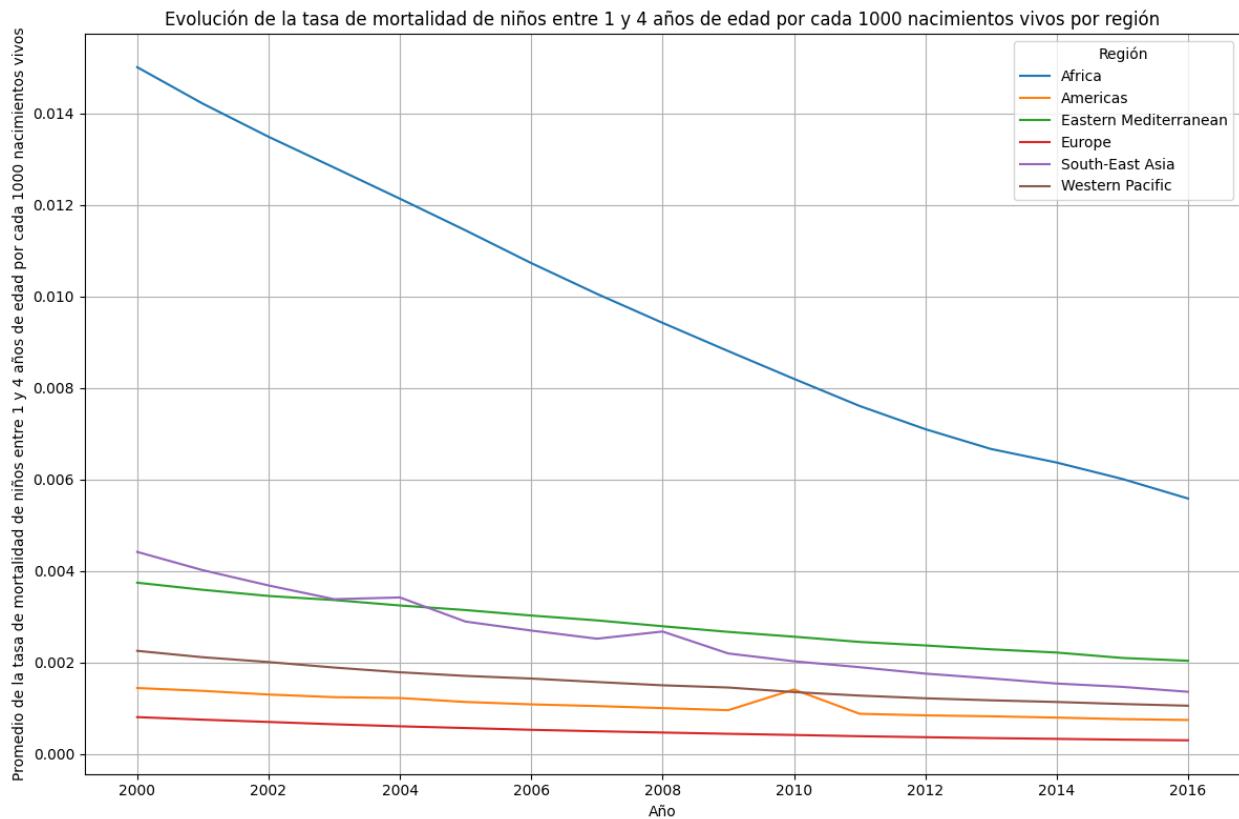


Evolución de la tasa de mortalidad de adultos entre los 15 y 60 años por cada 1000 habitantes por región

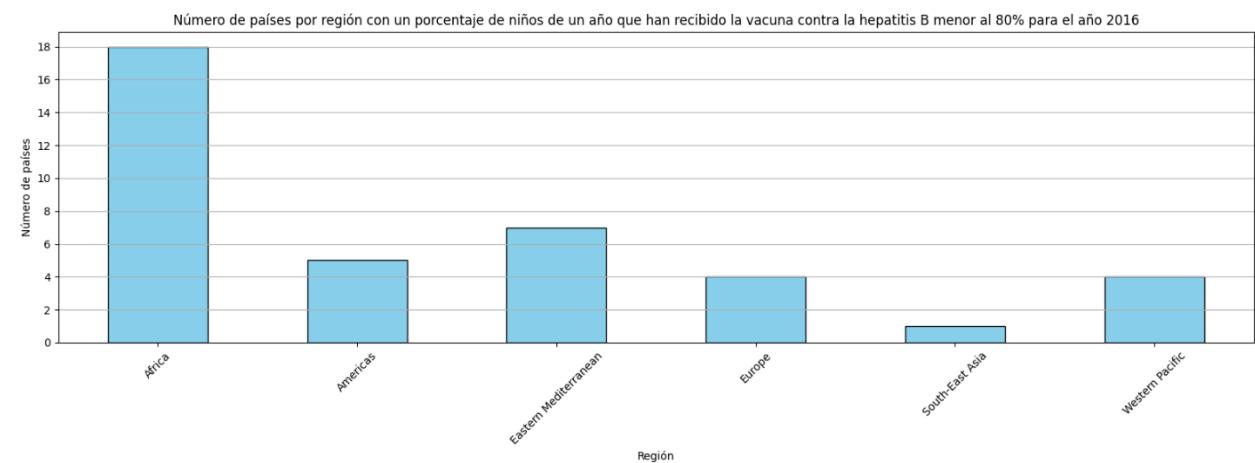
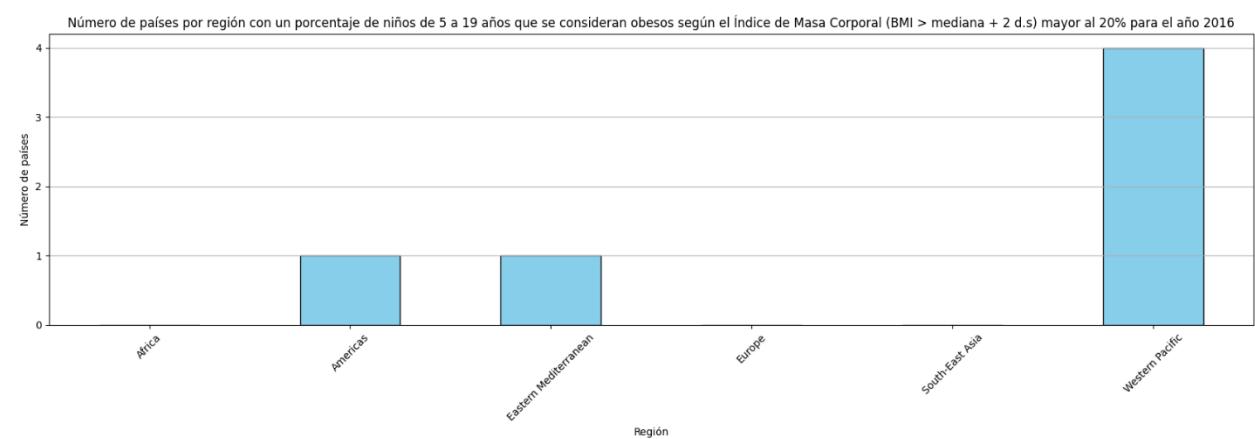
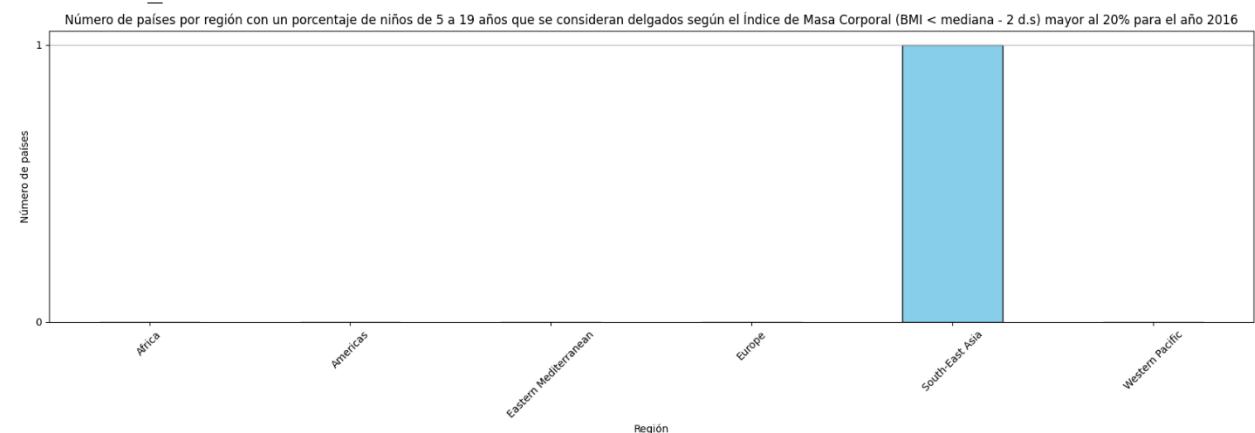


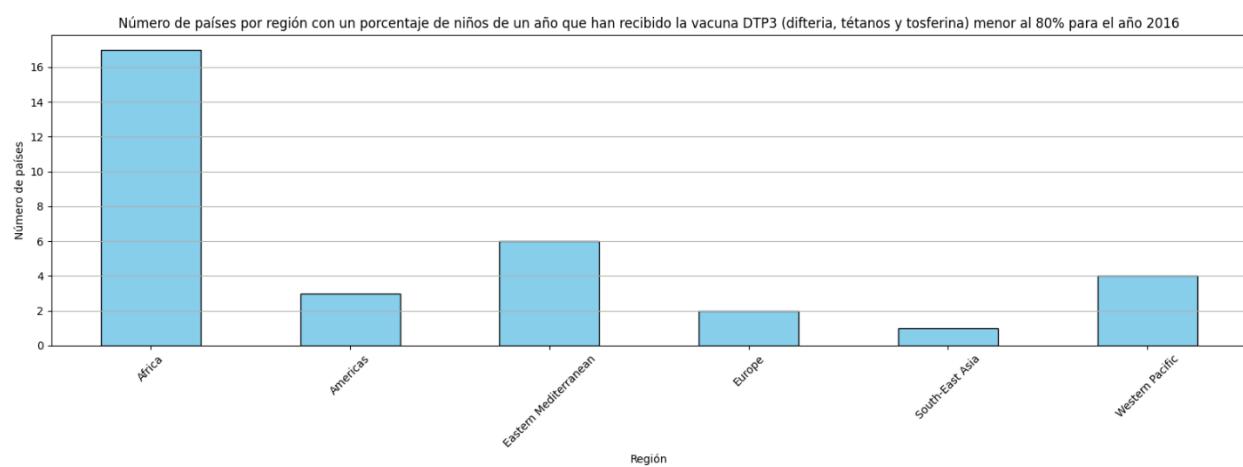
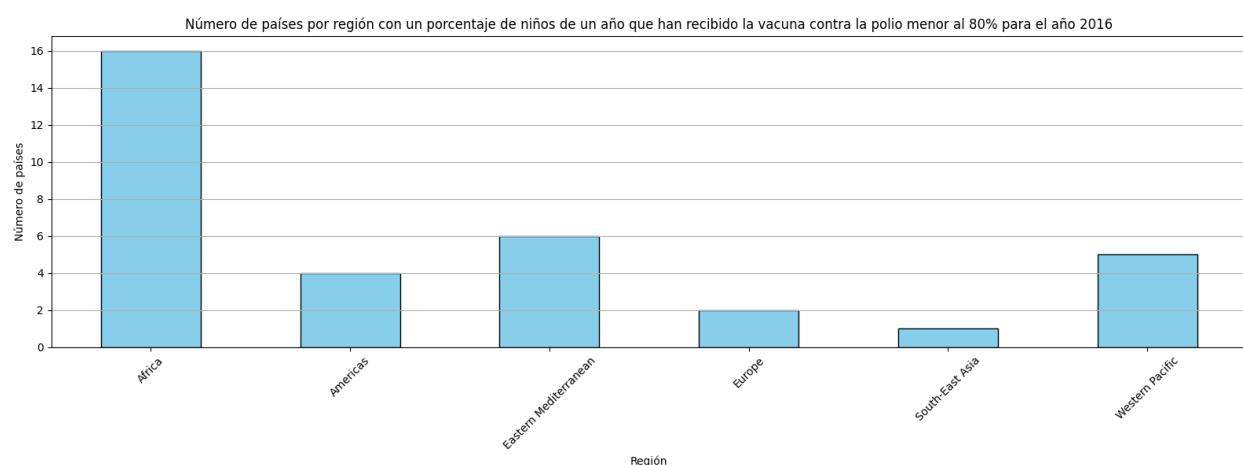
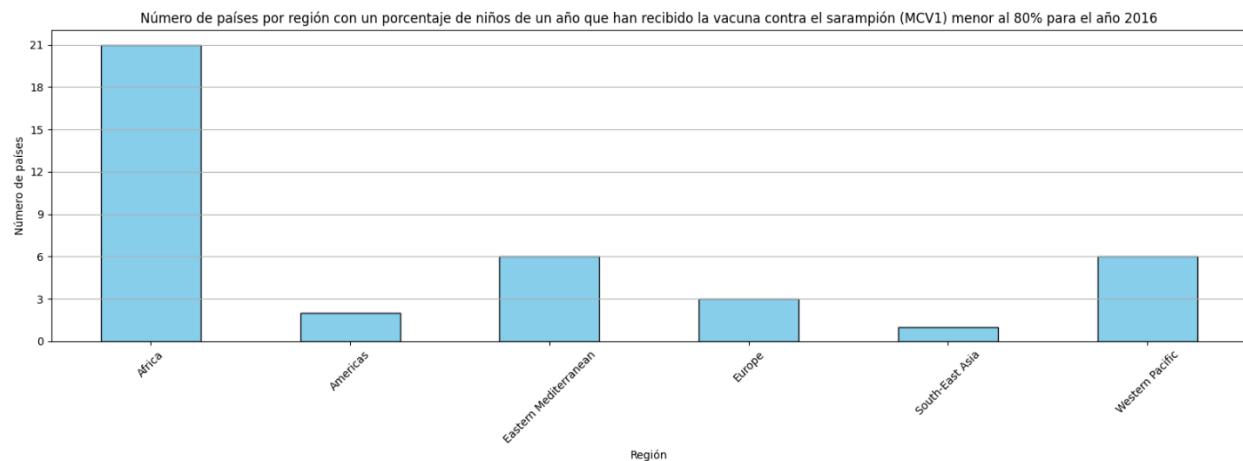
Evolución de la tasa de mortalidad de niños menores de un año por cada 1000 nacimientos vivos por región

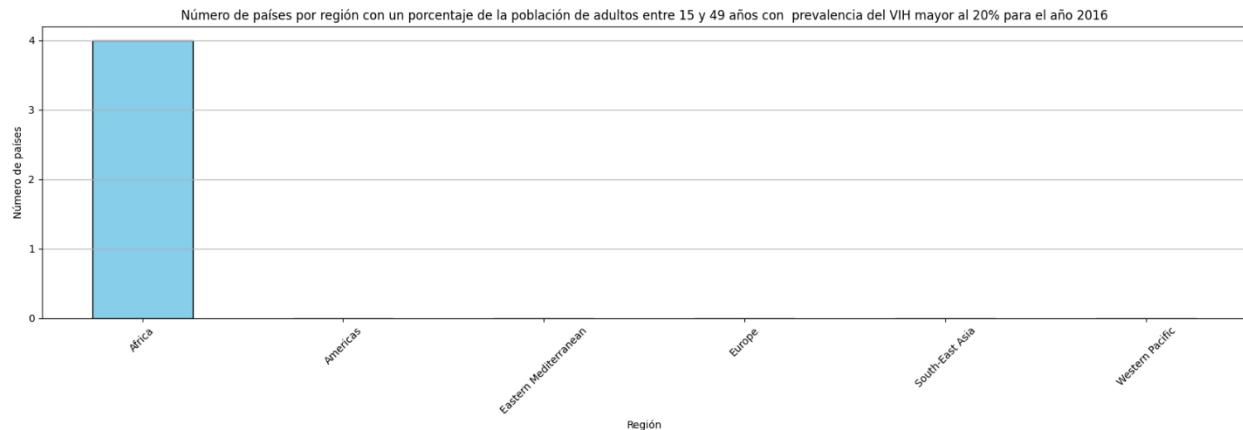




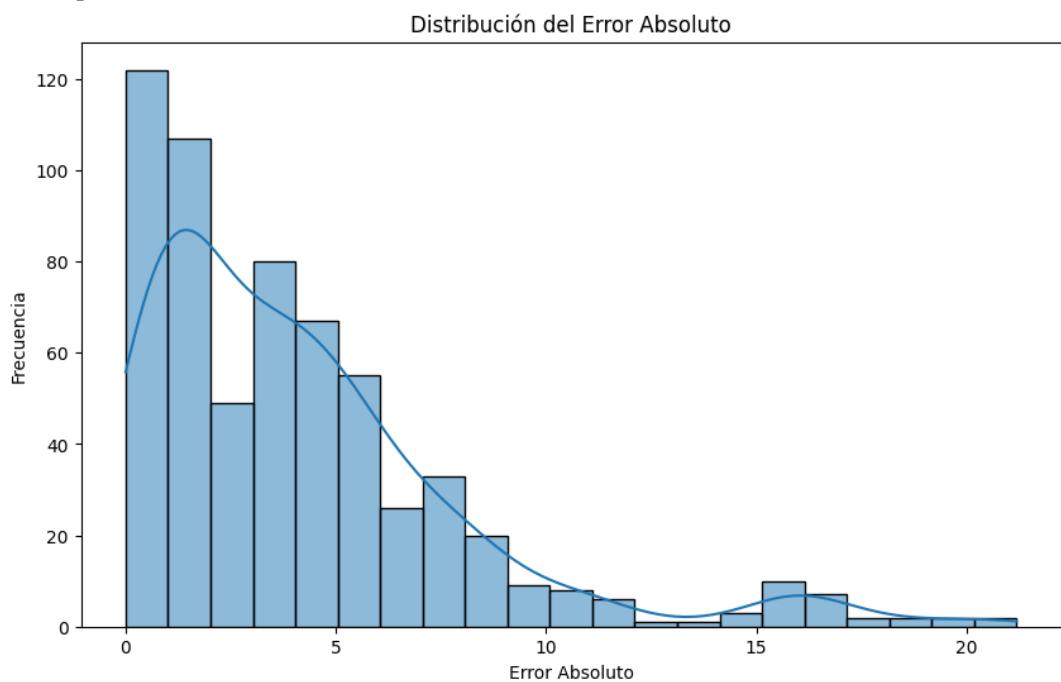
Para “*data_selected*”:



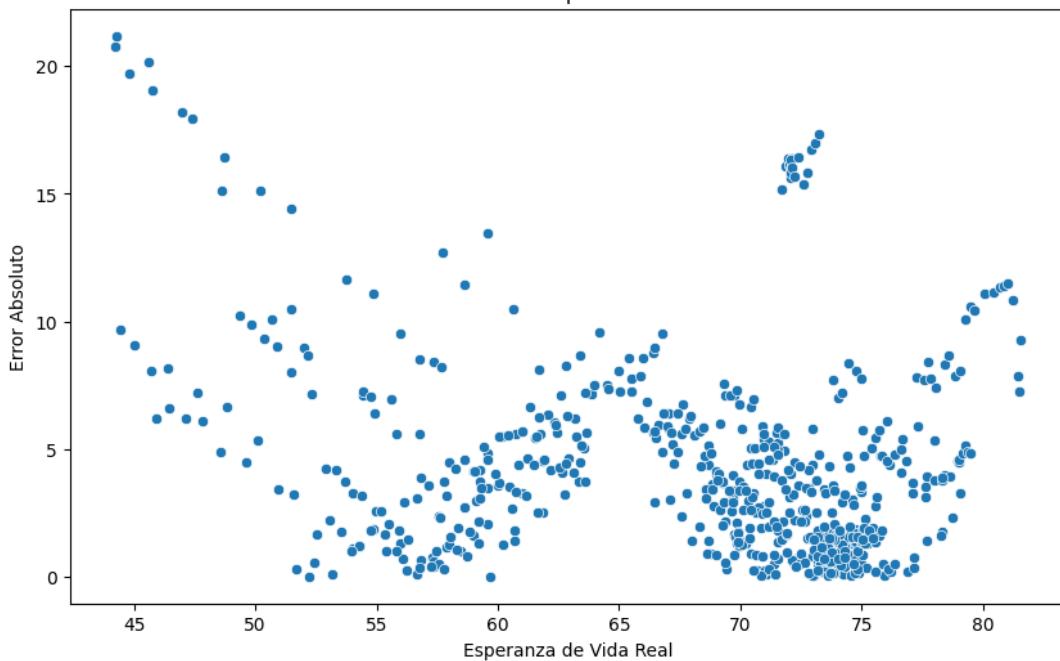




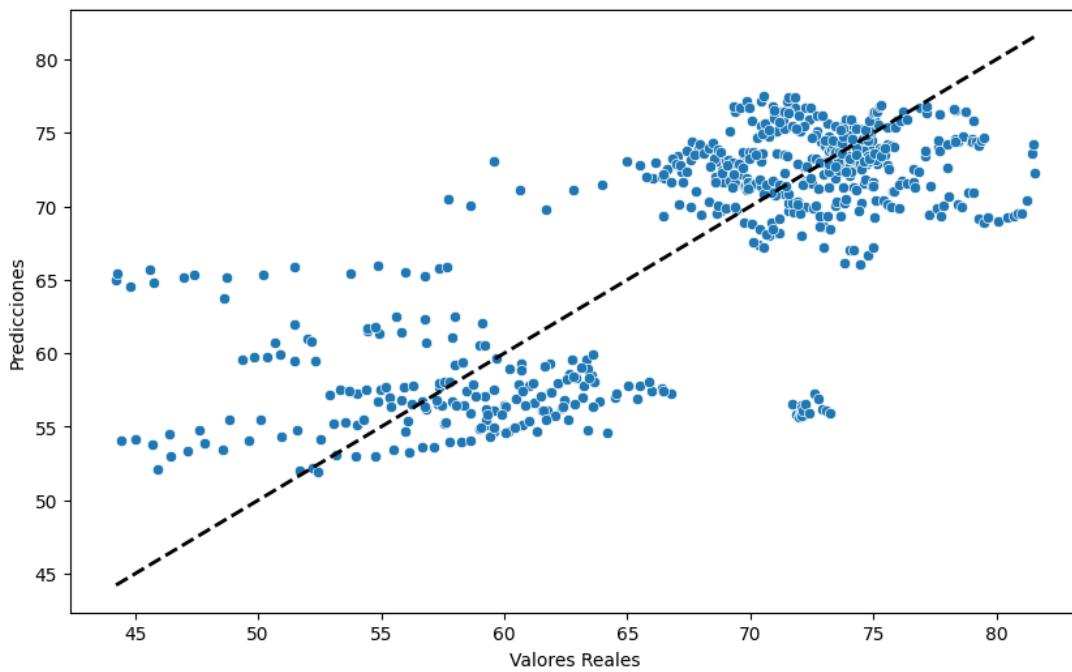
Para el desempeño del modelo:

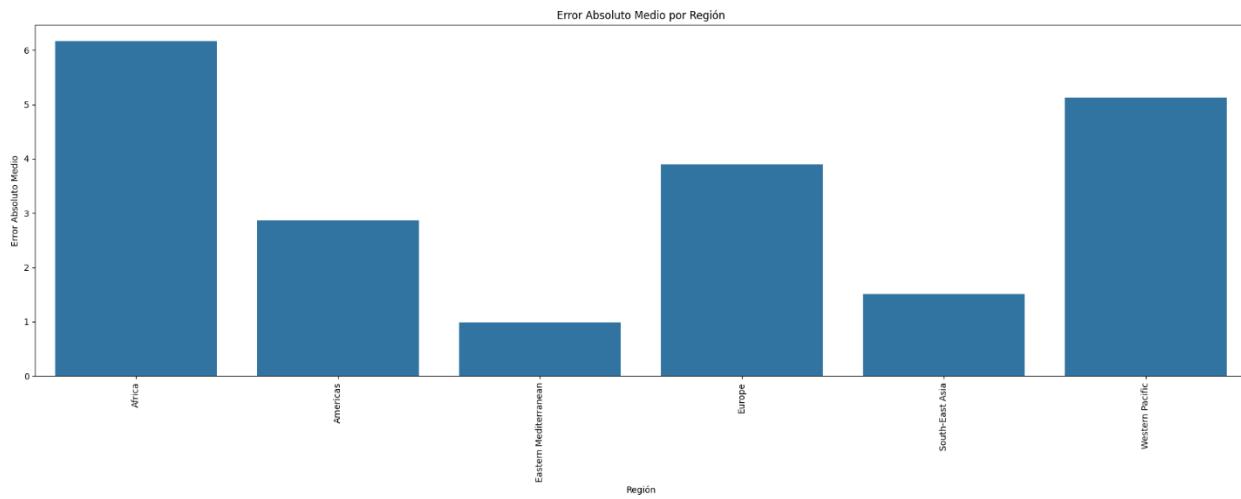
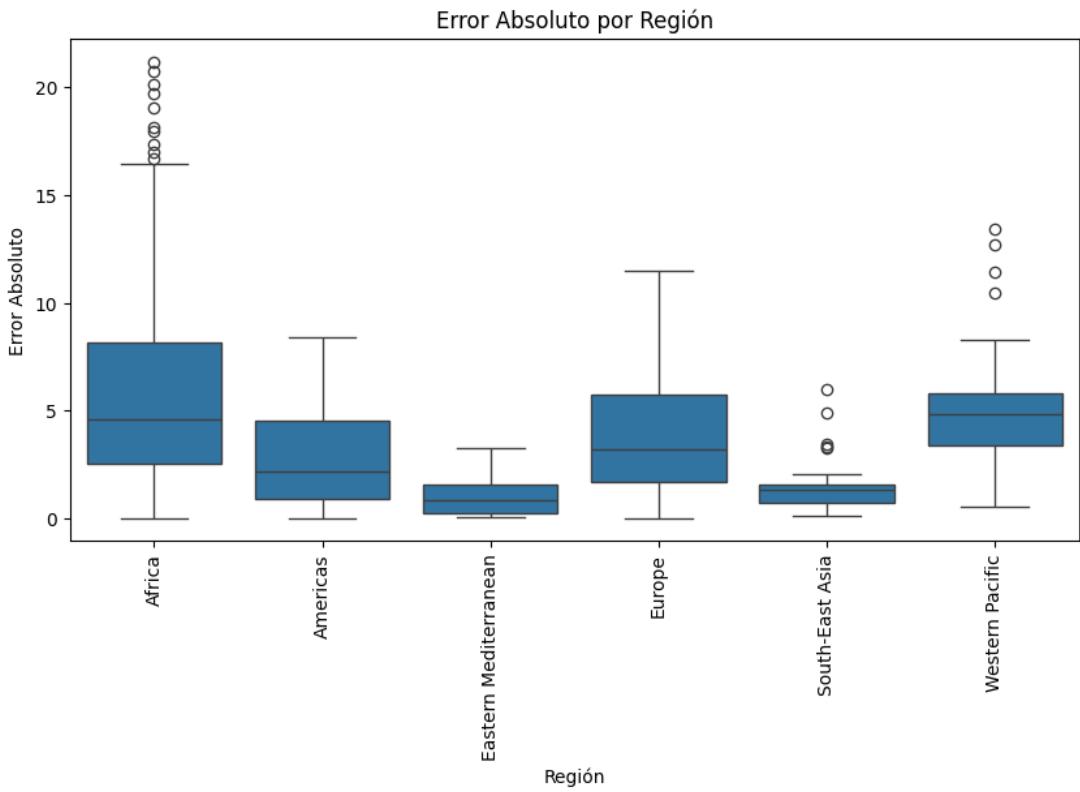


Error Absoluto vs. Esperanza de Vida Real



Predicciones vs. Valores Reales





Comparación entre implementaciones

El desempeño de ambos modelos difirió considerablemente. Mientras que el modelo entrenado con SparkML (con las particularidades de la imputación y la diferencia en parámetros), obtuvo los siguientes resultados:

MSE: 32.754939898875115

MAE: 4.199028801678624

El modelo entrenado con Scikit-learn, se desempeñó de la siguiente manera:

$$\text{MSE: } 9.371245996132622$$

$$\text{MAE: } 2.307437007489104$$

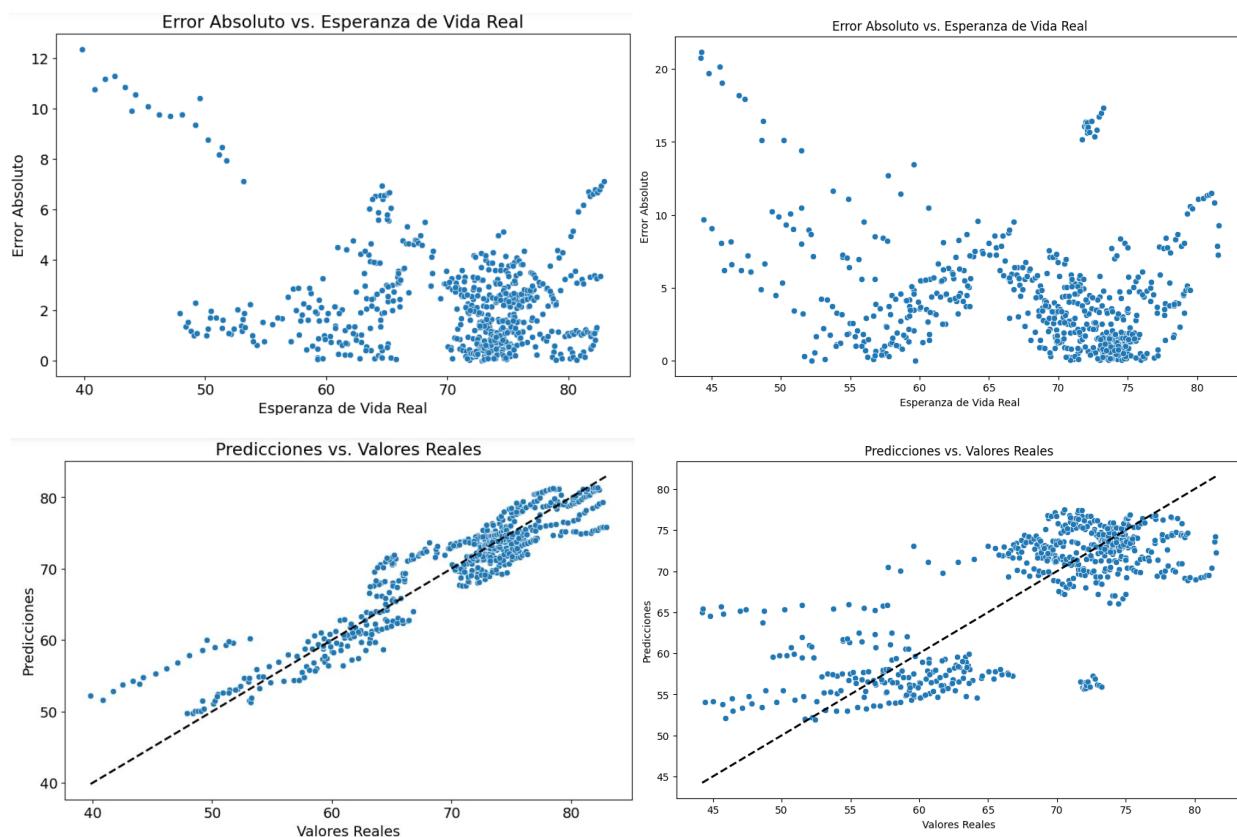
Por otra parte, el tiempo de entrenamiento de SparkML fueron **34.58 segundos**, el de Scikit-learn fueron **12.17 segundos**.

Tiempo de entrenamiento del modelo: 34.58 segundos

Tiempo de entrenamiento del modelo: 12.76 segundos

Tanto en el desempeño como en el tiempo de entrenamiento se aprecian diferencias considerables. Esto se debe, en primera instancia, a las funciones disponibles por cada librería en particular, y en segunda, al caso de uso (un conjunto de datos pequeño procesado con herramientas de Big Data).

De igual forma, se aprecia una mayor dispersión en los gráficos de error absoluto vs esperanza de vida real y predicciones vs valores reales, como se muestra a continuación (el gráfico de la izquierda corresponde al modelo entrenado con Scikit-learn, el de la derecha con SparkML):



Considerando los costos y complejidad de implementación del caso de uso de Big Data, y los resultados deficientes que derivaron, resulta evidente que esta no es optima para cumplir con los objetivos de negocio del proyecto.

Conclusiones

Conclusiones generales:

- El objetivo principal, junto a los objetivos específicos, fueron satisfechos.
- Se dio respuesta a la pregunta de investigación.
- El MAE del modelo construido en el demo de implementación en ambiente productivo con arquitectura batch, empleando servicios de Big Data de AWS, fue 1.82 veces mayor al construido en la implementación original.
- De la información derivada de este proyecto, se pueden formular juicios bien fundamentados que sustenten proyectos gubernamentales respaldados por la OMS para aumentar la longevidad y el bienestar de las poblaciones.

Posibles mejoras del modelo:

1. Aumentar la cantidad de variables significativas para la predicción de la esperanza de vida.
2. Aumentar el total de registros al incluir los datos de los últimos 7 años.
3. Distribución proporcional de las regiones en los conjuntos de entrenamiento, validación y prueba (eliminar distribución aleatoria).
4. Eliminar “Africa” como variable para impedir sesgo (habitar en Africa no es causal de menor Esperanza de vida).
5. Obtener los datos faltantes de los países por medios oficiales para reducir la proporción de datos imputados.
6. Entrenar otros modelos de ML como redes neuronales.
7. Utilizar la herramienta GridSearchCV para afinar los hiperparámetros del modelo y obtener el mejor rendimiento posible.

Reconocimientos

Durante todo el proyecto, se empleó el modelo de lenguaje natural Chat-GPT versión 4o para apoyar la producción de código y generación de conocimiento. Para todo caso de uso, la respuesta fue evaluada, comprobada, y comparada con los conocimientos propios y otras fuentes académicas, buscando siempre reducir al máximo la información imprecisa que pudo haber suministrado.

Referencias

Alanazi, A. (2022). Using machine learning for healthcare challenges and opportunities. *Informatics in Medicine Unlocked*, 30(100924), 100924. <https://doi.org/10.1016/j.imu.2022.100924>.

Geeks for Geeks. (2 de febrero de 2024). *Pandas DataFrame interpolate() Method | Pandas Method*. geeksforgeeks.org. <https://www.geeksforgeeks.org/pandas-dataframe-interpolate/>.

Geeks for Geeks. (4 de agosto de 2022). *PySpark Window Functions*. geeksforgeeks.org. <https://www.geeksforgeeks.org/pyspark-window-functions/>.

Graunt, J. (1662). *Natural and political observations made upon the bills of mortality*. Roycroft.

Halley, E. (1693). An estimate of the degrees of the mortality of mankind. *Philosophical Transactions of the Royal Society of London*, 17, 596–610. <https://doi.org/10.1098/rstl.1693.0007>.

Kontis, V., Bennett, J. E., Mathers, C. D., Li, G., Foreman, K., & Ezzati, M. (2017). Future life expectancy in 35 industrialised countries: Projections with a Bayesian model ensemble. *The Lancet*, 389(10076), 1323–1335. [https://doi.org/10.1016/S0140-6736\(16\)32381-9](https://doi.org/10.1016/S0140-6736(16)32381-9).

Lee, E. (5 de junio de 2019). *An Intro to Hyper-parameter Optimization using Grid Search and Random Search*. Medium. <https://medium.com/@cjlfv/an-intro-to-hyper-parameter-optimization-using-grid-search-and-random-search-d73b9834ca0a>.

Lipesa, B. A., Okango, E., Omolo, B. O., & Omondi, E. O. (2023). An application of a supervised machine learning model for predicting life expectancy. *SN Applied Sciences*, 5(7). <https://doi.org/10.1007/s42452-023-05404-w>.

López, J. F. (3 de octubre de 2017). *Teorema de Gauss-Márkov*. Economipedia.com. <https://economipedia.com/definiciones/teorema-gauss-markov.html>.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30. https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

Markus, A. F., Kors, J. A., & Rijnbeek, P. R. (2021). The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies. *Journal of Biomedical Informatics*, 113(103655), 103655. <https://doi.org/10.1016/j.jbi.2020.103655>.

Marmot, M. (2005). Social determinants of health inequalities. *The Lancet*, 365(9464), 1099–1104. [https://doi.org/10.1016/S0140-6736\(05\)71146-6](https://doi.org/10.1016/S0140-6736(05)71146-6).

Marmot, M. (2017). *The health gap: The challenge of an unequal world*. Bloomsbury Publishing.

McKeown, T. (1976). *The role of medicine: Dream, mirage, or nemesis?* Nuffield Provincial Hospitals Trust.

Organización Mundial de la Salud. (2024). *Esperanza de vida al nacer (años)*. WHO. <https://data.who.int/es/indicators/i/A21CFC2/90E2E48>.

Organización Mundial de la Salud. (2025). *Acerca de la OMS*. OMS. <https://www.who.int/es/about>.

Pooja, B., & Archana, U. (2023.). *Life Expectancy Prediction using Machine Learning*. Ijrpr.com. Recuperado el 1 de mayo de 2025, de <https://ijrpr.com/uploads/V4ISSUE12/IJRPR20743.pdf>.

Rodó, P. (18 de junio, 2019). *Multicolinealidad*. Economipedia.com. <https://economipedia.com/definiciones/multicolinealidad.html#:~:text=La%20multicolinealidad%20es%20la%20relaci%C3%B3n,m%C3%A1s%20de%20dos%20variables%20explicativas>.

United Nations Development Programme. (2025). *Human Development Index (HDI)*. Human Development Reports. Recuperado el 1 de mayo de 2025, de <https://hdr.undp.org/data-center/human-development-index#/indicies/HDI>.

Wang, H., et al. (2016). Global, regional, and national life expectancy, all-cause mortality, and cause-specific mortality for 249 causes of death, 1980–2015: A systematic analysis for the Global Burden of Disease Study 2015. *The Lancet*, 388(10053), 1459–1544. [https://doi.org/10.1016/S0140-6736\(16\)31012-1](https://doi.org/10.1016/S0140-6736(16)31012-1).

World Health Organization. (2023). *World health statistics 2023: Monitoring health for the SDGs*. WHO. <https://www.who.int/data/gho/publications/world-health-statistics>.