Estimación de los niveles de obesidad en función de los hábitos alimenticios y la condición física para una población en el año 2019.

Un ánalisis con Python.

Sofia Arias Juárez
Escuela de informática

Universidad Nacional de Costa Rica
Heredia, Costa Rica
https://orcid.org/0009-0008-4095-5553

Luis Chavarría Chacón
Escuela de informática

Universidad Nacional de Costa Rica
Heredia, Costa Rica
https://orcid.org/0009-0008-1663-9024

Andrés González Romero
Escuela de informática

Universidad Nacional de Costa Rica
Heredia, Costa Rica
https://orcid.org/0009-0002-7691-643X

Alisson Steller Alfaro
Escuela de informática

Universidad Nacional de Costa Rica
Heredia, Costa Rica
https://orcid.org/0009-0009-9689-489X

***Abstract*— La siguiente investigación permite dar a conocer el análisis de los niveles de obesidad según hábitos alimenticios y la condición física de las personas de Colombia, México y Perú para el año 2019, para lo cual se empleará el uso de librerías Python para el análisis de datos, como lo son Pandas y Matplotlib. Con el fin de analizar cómo los patrones de alimentación, la actividad física, y otros hábitos de estilo de vida impactan en el estado de obesidad de los individuos.**

Keywords—Python, Análisis de datos, obesidad, Pandas.

I. INTRODUCCIÓN

La obesidad es un problema de salud pública global que ha experimentado un aumento preocupante en las últimas décadas. Afecta a millones de personas en todo el mundo y está asociada con numerosas enfermedades crónicas, incluidas las enfermedades cardiovasculares, diabetes tipo 2 y ciertos tipos de cáncer. Este estudio se enfoca en la estimación de los niveles de obesidad basada en los hábitos alimenticios y la condición física de individuos de Colombia, México y Perú en el año 2019, utilizando funciones de análisis de datos con herramientas de Python.

Por ende, se aprovechará las capacidades de librerías como Pandas, para la manipulación y evaluación de datos, y Matplotlib, para la visualización de los resultados. A través del análisis de datos recolectados, se busca identificar patrones y correlaciones entre la alimentación, la actividad física y los niveles de obesidad. La forma en que se utilizaron las herramientas de esta investigación será explicada conforme se vaya detallando cada punto de la misma.

El objetivo de esta investigación es analizar cómo los patrones de alimentación, la actividad física, y otros hábitos de estilo de vida impactan en el estado de obesidad de los individuos. Además, se explorará la importancia de utilizar herramientas de análisis de datos que están en la capacidad de procesar grandes cantidades de datos para llegar a conclusiones que puedan utilizarse para ejecutar planes de acción que ayuden a mitigar la obesidad en las personas. Este estudio, también tiene la intención de contribuir al conocimiento global sobre la obesidad, ofreciendo perspectivas que pueden ser aplicadas en otros contextos, además del desarrollo de programas de salud orientados a combatir esta enfermedad en América Latina y otras partes del mundo.

II. MARCO TEÓRICO

A continuación, se van a describir 10 casos de éxito en el mundo acerca de Python para demostrar la gran potencialidad que tiene el uso de librerías en este mismo lenguaje.

A. Cooperación escuela-empresa en la enseñanza del análisis de datos con Python

Este artículo se realizó en China. Plantea una metodología para enseñar el análisis de datos usando el lenguaje de programación de Python. Dentro del mismo se menciona el diseño de un curso con el propósito de desarrollar talentos en los estudiantes, utilizando como base casos prácticos dados según la demanda de las empresas.

De acuerdo con [1], Python fue empleado fundamentalmente para la adquisición y procesamiento de datos, incluyendo formatos como CSV, bases de datos y API. Este enfoque se apoyó en el uso de librerías especializadas como Pandas, que facilita la manipulación y análisis de datos. Para los análisis estadísticos más complejos, se recurrió a librerías como Numpy, SciPy y scikit-learn, las cuales son esenciales para realizar cálculos numéricos y científicos avanzados, permitiendo a los estudiantes explorar y aplicar técnicas estadísticas y de modelado de datos de manera efectiva.[1]

B. Python para el análisis de datos y aplicaciones científicas y técnicas

El estudio se llevó a cabo en la India. Su objetivo principal fue analizar por qué Python ha ganado tanta popularidad como

lenguaje de programación en aplicaciones científicas, técnicas y de análisis de datos. Los investigadores se enfocaron en las características distintivas de Python, como su facilidad de aprendizaje, su flexibilidad y su creciente conjunto de bibliotecas, que lo convierten en una solución integral para desarrollo en campos como ciencia de datos, inteligencia artificial y aprendizaje profundo.

En cuanto a la aplicación de Python, este se utilizó para desarrollar métodos computacionales avanzados y obtener resultados valiosos de los datos mediante herramientas de visualización. Python también facilitó la programación de algoritmos complejos en áreas como el aprendizaje automático y el procesamiento del lenguaje natural, utilizando bibliotecas como TensorFlow y NLTK. Además, permitió la integración de software heredado y la optimización de la eficiencia mediante la programación paralela y la vectorización en algoritmos, demostrando ser una herramienta esencial para la codificación científica efectiva. [2]

C. Análisis exploratorio de datos para interpretar la predicción del modelo utilizando Python

Esta investigación se realizó en Mohali, India. Su objetivo principal fue explorar y demostrar la eficacia de las técnicas de Análisis de Datos Exploratorios (EDA) y de aprendizaje automático, utilizando el lenguaje de programación Python para optimizar e interpretar predicciones de modelos basados en datos. Se centró en el uso del conjunto de datos Iris para entrenar y optimizar un modelo predictivo que pudiera realizar futuras predicciones con una notable precisión.

En cuanto a la aplicación de Python, se utilizó como la principal herramienta para el análisis exploratorio de datos y el entrenamiento de modelos de aprendizaje automático. Se emplearon bibliotecas de Python como Numpy para operaciones matemáticas con arrays, Pandas para la manipulación de datos, Matplotlib y Seaborn para visualizaciones, y Scikit-Learn para implementar algoritmos de aprendizaje automático. Estas bibliotecas facilitaron significativamente la manipulación de datos, la visualización gráfica y la implementación de modelos predictivos, permitiendo evaluar y mejorar la precisión de los modelos en un entorno controlado y efectivo.[3]

D. PyOMP: Programación paralela multihilo en Python

El estudio se llevó a cabo en Hillsboro, Oregon, y en Livermore, California, EE.UU. El objetivo fue desarrollar PyOMP, un sistema que habilita la programación paralela multihilo en Python utilizando OpenMP. El propósito principal era superar las limitaciones de rendimiento de Python en comparación con lenguajes de bajo nivel como C y Fortran, especialmente en aplicaciones que exigen alto rendimiento.

Este logró implementar y demostrar que el código Python, cuando se ejecuta con PyOMP, puede alcanzar velocidades de ejecución comparables a las de programas similares escritos en C con OpenMP. Python fue utilizado con la ayuda de Numba, un compilador Just-In-Time (JIT) que traduce Python a código LLVM, permitiendo así la compilación y ejecución eficiente. PyOMP se implementó de manera que permita el uso de directivas OpenMP, facilitando la escritura de código multihilo paralelo que es tanto eficiente como fácil de entender para los programadores de Python. [4]

E. Automatización del análisis de datos con Python: Un estudio comparativo de bibliotecas populares y su aplicación

Se realizó una comparativa exhaustiva de varias librerías populares en Python dedicadas a la automatización del análisis de datos. Como resultado, se identificó que librerías como Pandas y NumPy demuestran una alta eficiencia en tareas de preprocesamiento y limpieza de datos debido a su capacidad para manejar y manipular grandes conjuntos de datos de manera efectiva. Para tareas más complejas de aprendizaje automático y aprendizaje profundo, librerías como TensorFlow y Scikit-learn resultaron ser más adecuadas debido a su especialización en algoritmos de machine learning y su capacidad para entrenar modelos de manera eficiente.

Durante la investigación, se utilizaron grandes datasets para poner a prueba estas librerías, realizando una serie de evaluaciones que permitieron comparar su rendimiento. Las pruebas se centraron en medir la eficiencia computacional, la facilidad de uso y la precisión en la ejecución de tareas específicas de análisis de datos.[5]

F. Análisis de datos financieros y cuantificación de riesgos basados en Python

Este artículo recalca la importancia de ciertas bibliotecas del lenguaje de Python como lo son Pandas y Numpy para el análisis de datos financieros. Pandas ofrece una gran capacidad de procesar, analizar y organizar datos de una forma veloz y sencilla. Por otro lado, Numpy es un paquete que posee una variedad de funciones para realizar operaciones complejas con arrays y matrices, siendo estas más eficientes en comparación a utilizar solamente el lenguaje de Python para realizarlas.

Python se aplicó en varios aspectos del análisis financiero. Se utilizó para recopilar datos a través de bibliotecas como Tushare y AkShare, procesar y analizar estos datos utilizando NumPy y Pandas, y visualizar los resultados mediante Matplotlib y Seaborn. Además, Python facilitó la implementación de modelos estadísticos y algoritmos de aprendizaje automático para la evaluación de riesgos y la predicción de comportamientos financieros. El estudio también hizo uso de la distribución Anaconda de Python para simplificar la gestión del entorno de desarrollo y las dependencias de las bibliotecas utilizadas.[6]

G. Análisis de Big Data de la popularidad del índice de vídeo y la emoción del público basado en Python

El estudio fue hecho en Pengshan, Sichuan, China. El objetivo principal consistió en analizar la popularidad de los índices de videos y la opinión pública utilizando tecnologías de big data basadas en el lenguaje de programación Python. Esta investigación se centró en capturar y analizar datos sobre la popularidad del contenido y las tendencias de las opiniones expresadas en los comentarios, entre otros indicadores clave, para comprender mejor la recepción e impacto de los videos en las plataformas en línea.

Utilizando técnicas de web scraping y análisis de datos, el equipo investigador pudo identificar tendencias significativas y obtener hallazgos profundos sobre la percepción pública del contenido en video. Python se utilizó extensivamente a lo largo del estudio, aplicando tecnología de web crawler para recolectar datos de sitios web de índices de video como el Qiyi Index. Posteriormente, se emplearon bibliotecas de Python para el análisis y la visualización de estos datos, facilitando la interpretación de tendencias complejas y la evaluación de la popularidad y las opiniones públicas. [7]

H. *Web Scraping for Data Analytics: Una implementación de BeautifulSoup*

Este análisis se llevó a cabo en Riyadh, Arabia Saudita. Su objetivo principal fue desarrollar un web scraper utilizando la biblioteca BeautifulSoup de Python para recoger automáticamente datos de cualquier sitio web y analizarlos posteriormente.

Python se aplicó utilizando su biblioteca BeautifulSoup para navegar y analizar el contenido HTML de las páginas de Amazon, extrayendo datos específicos como nombre del producto, precio y reseñas. Para la visualización de los datos recolectados, se utilizó la biblioteca Matplotlib para generar gráficos que facilitaron la interpretación de los datos, tales como la frecuencia de precios y el número de reseñas por calificación del producto. [8]

I. *Análisis de datos de mezcla de combustibles de generación de suministro eléctrico en línea utilizando Python y TensorFlow.*

Otra aplicación importante de Python en las investigaciones fue la extracción de los datos de un código fuente HTML, dividiéndolos en varios archivos de este y exportarlos en el formato CSV. Todo lo anterior, para crear un sistema software a partir aprendizaje automático de la inteligencia artificial, que pudiera extraer, visualizar y analizar la información relacionada con la mezcla de combustible utilizada por los proveedores de electricidad en el mercado energético del Reino Unido,

El sistema realizó análisis automatizados de tendencias en el suministro de electricidad, para su visualización se hace uso de librerías como: Matplotlib y Seaborn, con el fin de proporcionar al usuario una fácil representación de los datos, por otro lado, Python se aplicó como el lenguaje principal de programación en el desarrollo del sistema de software, junto con Tensorflow para el aprendizaje automático de la IA. [9]

J. *Sistema de minería de datos educativos y análisis del aprendizaje basado en Python*

Para concluir, una de las investigaciones utilizando Python en el análisis de datos, tuvo lugar en China. El propósito de la investigación fue desarrollar un sistema sobre el aprendizaje de los estudiantes, tomando como base la información recolectada de las plataformas educativas, las cuales ellos usan, con el fin de proporcionar un aprendizaje personalizado y por ende obtener mejores resultados académicos.

Se buscó un enfoque el cual se utilizaron técnicas de minería de datos, entre ellas, una variedad de métodos estadísticos y algoritmos para extraer información valiosa de información de las plataformas de educación en línea. Para lograr lo anterior, Python utilizó librerías como: Scikit-Learn y Keras, y fue el lenguaje principal para el desarrollo del sistema en su totalidad, con tareas tales: creación de interfaz, aplicación de algoritmos de aprendizaje automático e integración con las bases de datos. [10]

III. Descripción de Página

La información de este conjunto de datos se obtuvo de la página web del Repositorio de Aprendizaje Automático de la Universidad de California, Irvine (UCI) para el primer semestre de 2024. Es una plataforma reconocida por proporcionar datos abiertos valiosos para la comunidad científica y académica interesada en el análisis estadístico y el aprendizaje automático.

Entre la diversidad de información disponible en este sitio, se seleccionó una base de datos con 2111 registros del año 2019, la cual tiene datos de los siguientes tres países: Colombia, México y Perú, que aborda la estimación de los niveles de obesidad para destacar cómo los patrones de alimentación y ejercicio afectan la salud de las personas. Esta Base de Datos incluye 17 variables, entre las cuales se encuentran el género, la edad, altura, peso, historial familiar con sobrepeso, consumo de alimentos con altas calorías, consumo de vegetales, cantidad de comidas principales al día, consumo de alguna comida entre las comidas principales, consumo de tabaco, cantidad de agua consumida diariamente, monitoreo de cantidad de calorías consumidas diariamente, frecuencia de actividad física, uso de dispositivos tecnológicos, consumo de alcohol, medio de transporte utilizado, y nivel de obesidad. Para cada uno de estos atributos se proporciona una breve descripción, lo que ayuda a los usuarios a entender y utilizar los datos de manera efectiva en sus investigaciones, permitiendo una evaluación detallada y multidimensional de los factores que contribuyen a la obesidad.

Se podrá acceder a estos registros mediante el siguiente enlace de descarga.

IV. Descripción De Procesamiento de la información

Para realizar el procesamiento de la información se hizo uso de Python con Google Colab para poder analizar datos y crear gráficas. Se hizo uso de la librería Pandas, con esta se procesaron los datos del archivo llamado “Obesity.csv”, a su vez se utilizó Matplotlib para diseñar los gráficos.

A. *Nivel de obesidad por género (cantidad hombres y cantidad mujeres)*

Para la solución, se utilizó la librería Pandas en Python para procesar y analizar los datos de obesidad. Primero, se cargó el archivo CSV que contiene la información relevante. Luego, se filtraron los datos para incluir solo los registros con información sobre el género (hombres y mujeres) y los niveles

de obesidad. Posteriormente, se agruparon los datos por nivel de obesidad y género, y se contó el número de registros en cada grupo. Esto se logró haciendo uso de las funciones groupby y size de Pandas, y luego se reorganizaron los resultados con unstack para obtener un DataFrame con columnas separadas para el número de hombres y mujeres en cada nivel de obesidad. Finalmente, se renombraron las columnas para mayor claridad y se visualiza la tabla resultante, la cual proporciona una visión clara de la distribución de los niveles de obesidad por género.

```
 1. import pandas as pd
 2.
 3. df = pd.read_csv('Obesity.csv')
 4.
 5. registro_general =
df.loc[(df['Gender'].isin(['Male', 'Female'])) &
(df['NObeyesdad'].notnull())]
 6.
 7. conteo_obesidad_genero =
registro_general.groupby(['NObeyesdad',
'Gender']).size().unstack(fill_value=0)
 8.
 9. conteo_obesidad_genero.columns = ['Número de
Mujeres', 'Número de Hombres']
10.
11. print("Nivel de Obesidad por Género:")
12. print(conteo_obesidad_genero)
13.
```

Resultado:

Nivel de Obesidad por Género:

NObeyesdad	Número de Mujeres	Número de Hombres
Insufficient_Weight	173	99
Normal_Weight	141	146
Obesity_Type_I	156	195
Obesity_Type_II	2	295
Obesity_Type_III	323	1
Overweight_Level_I	145	145
Overweight_Level_II	103	187

Tabla 1 Nivel de obesidad por género

Conclusión:

Estos resultados pueden sugerir diferentes patrones de obesidad y peso insuficiente entre hombres y mujeres. Es posible que haya factores biológicos, sociales, culturales o comportamentales que contribuyan a estas diferencias. Las mujeres parecen ser más propensas a alcanzar niveles extremos de obesidad (Obesidad Tipo III), mientras que los hombres tienen una mayor incidencia de obesidad severa (Obesidad Tipo II).

Estas observaciones podrían ser útiles para diseñar intervenciones de salud pública específicas para cada género, enfocándose en los diferentes desafíos que enfrentan hombres y mujeres en relación con la obesidad y el peso insuficiente.

B. Promedio del peso por nivel de obesidad y género

Para la solución de este problema se utilizó un enfoque similar al anterior, se hizo uso de la librería Pandas en Python. De igual manera se carga el archivo CSV que contiene la información relevante. Luego, se filtraron los datos para incluir solo los registros con información sobre el género (hombres y mujeres) y los niveles de obesidad. Posteriormente, se agruparon los datos por nivel de obesidad y género, y se calculó el promedio del peso en cada grupo utilizando la función **mean** de Pandas. Se reorganizaron los resultados con unstack para obtener un DataFrame con columnas separadas para el promedio de peso de hombres y mujeres en cada nivel de obesidad. Luego, se redondearon los resultados a dos decimales para mayor precisión. Finalmente, se renombraron las columnas para mayor claridad y se visualiza la tabla resultante.

```
 1. import pandas as pd
 2. df = pd.read_csv('Obesity.csv')
 3.
 4. registro_general =
df.loc[(df['Gender'].isin(['Male', 'Female'])) &
(df['NObeyesdad'].notnull())]
 5.
 6. promedio_peso_obesidad_genero =
registro_general.groupby(['NObeyesdad',
'Gender'])['Weight'].mean().unstack(fill_value=0)
 7.
 8. promedio_peso_obesidad_genero =
promedio_peso_obesidad_genero.round(2)
 9.
10. promedio_peso_obesidad_genero.columns =
['Promedio de Peso Mujeres', 'Promedio de Peso
Hombres']
11.
12. print("Promedio del Peso por Nivel de Obesidad y
Género:")
13. print(promedio_peso_obesidad_genero)
14.
```

Resultado:

Promedio del peso por nivel de obesidad y género:

NObeyesdad	Promedio de Peso Mujeres	Promedio de Peso Hombres
Insufficient_Weight	46.69	55.53
Normal_Weight	56.42	67.70
Obesity_Type_I	82.29	101.33
Obesity_Type_II	96.75	115.43
Obesity_Type_III	120.78	173.00
Overweight_Level_I	69.58	78.95
Overweight_Level_II	74.54	86.24

Tabla 2 Promedio del peso por nivel de obesidad y género

Conclusión:

Al calcular el promedio del peso por nivel de obesidad y género, se obtiene información que revela cómo varía el peso promedio entre los diferentes niveles de obesidad y géneros.

Este análisis permite identificar tendencias y diferencias significativas entre hombres y mujeres en relación con su peso y nivel de obesidad. Por ejemplo, se observa que las mujeres tienden a tener un peso promedio más bajo en la categoría de

"Insufficient Weight" en comparación con los hombres. En la categoría de "Normal Weight", los pesos promedio de hombres y mujeres son similares, lo que indica una distribución de peso más equilibrada. Sin embargo, en las categorías de obesidad, los hombres tienden a tener un peso promedio más alto en ciertas categorías como "Obesity_Type_I" y "Obesity_Type_II".

Este tipo de análisis es fundamental para la elaboración de intervenciones, abordando los distintos desafíos que hombres y mujeres enfrentan en relación con el peso y la obesidad. Además, estos datos pueden ser útiles para los profesionales de la salud al desarrollar planes de tratamiento personalizados y efectivos.

C. Promedio de edad por nivel de obesidad y género

Para esta solución, se hizo uso de la librería Pandas en Python. Primero, se cargó el archivo CSV que contiene la información relevante. Luego, se filtraron los datos para incluir solo los registros con información sobre el género (hombres y mujeres), el nivel de obesidad y la edad. Posteriormente, haciendo uso de las funciones de **.groupby()** y **.mean()**, se agruparon los datos por nivel de obesidad y género, y se calculó el promedio de edad en cada grupo. Finalmente, se visualizó la tabla resultante, que proporciona una visión clara del promedio de edad por nivel de obesidad y género.

```
 1. import pandas as pd
 2.
 3. df = pd.read_csv('Obesity.csv')
 4.
 5. registro_general=
 6. df.loc[(df['Gender'].isin(['Male','Female']))&
(df['NObeyesdad'].notnull()) &
 7. (df['Age'].notnull())]
 8.
 9.
promedio_edad_obesidad_genero=registro_general.groupb
y(['NObeyesdad',
'Gender'])['Age'].mean().unstack(fill_value=0)
10.
11. promedio_edad_obesidad_genero.columns =
['Promedio de Edad Mujeres', 'Promedio de Edad
Hombres']
12.
13. print("Promedio de Edad por Nivel de Obesidad y
Género:")
14. print(promedio_edad_obesidad_genero)
15.
```

Resultado:

Promedio de edad por nivel de obesidad y género

NObeyesdad	Promedio de Edad Mujeres	Promedio de Edad Hombres
Insufficient_Weight	20.50	18.51
Normal_Weight	22.02	21.45
Obesity_Type_I	27.89	24.27
Obesity_Type_II	24.50	28.25
Obesity_Type_III	23.51	18.00
Overweight_Level_I	24.57	22.25
Overweight_Level_II	27.38	26.78

Tabla 3 promedio de edad por obesidad y género

Conclusión:

Estos resultados indican diferencias en el promedio de edad entre hombres y mujeres en diferentes niveles de obesidad. Esto sugiere que la edad podría estar relacionada con la prevalencia de ciertos niveles de obesidad y que estas relaciones pueden variar según el género. Estas observaciones podrían ser útiles para diseñar intervenciones de salud pública específicas, considerando la edad y género de las personas.

D. Porcentaje de los niveles de obesidad según registros con historial familiar de obesidad a nivel general

Con el fin de obtener el porcentaje de los niveles de obesidad según los registros con historial familiar de obesidad, se ha decidido hacer uso de la herramienta de pandas. Primeramente, se cargó un archivo de tipo CSV, luego a partir de este se contaron todos los registros que incluyesen la palabra **"yes"** dentro de la columna de **family_history_with_overweight**. Seguidamente, se calculó el porcentaje de los niveles de obesidad con base en el dato extraído anteriormente.

```
 1. import pandas as pd
 2.
 3. df = pd.read_csv('Obesity.csv')
 4.
 5. historial_familiar=
df[df['family_history_with_overweight'] == 'yes']
 6. total_historial_familiar =
len(historial_familiar)
 7. porcentaje_obesidad=
(historial_familiar['NObeyesdad'].value_counts() /
total_historial_familiar) * 100
 8. porcentaje_obesidad_df=
porcentaje_obesidad.reset_index()
 9. porcentaje_obesidad_df.columns = ['Nivel de
Obesidad', 'Porcentaje']
10.
11. print("Porcentaje de Niveles de Obesidad con
Historial Familiar de Obesidad:")
12. print(porcentaje_obesidad_df)
13.
14.
```

Resultado:

Porcentaje de los niveles de obesidad según registros con historial familiar de obesidad

Nivel de Obesidad	Porcentaje
Obesity_Type_I	19.93
Obesity_Type_III	18.77
Obesity_Type_II	17.14
Overweight_Level_II	15.75
Overweight_Level_I	12.10

Normal_Weight	8.98
Insufficient_Weight	7.30

Tabla 4 porcentaje de los niveles de obesidad según registros con historial familiar de obesidad

Conclusión:

Estos resultados permiten identificar cómo se distribuyen los niveles de obesidad entre las personas con historial familiar de obesidad, lo cual puede ser útil para diseñar intervenciones específicas y entender el impacto del historial familiar en la obesidad.

E. Porcentaje de los niveles de obesidad según registros sin historial familiar de obesidad

Para calcular el porcentaje de los niveles de obesidad según los registros sin historial familiar de obesidad, se utilizó la librería Pandas de Python. Primero, se importó dicha librería y se cargó el archivo CSV que contiene la información del estudio. Luego, se procedió a filtrar únicamente aquellas filas que es de interés. Para realizar este paso, se utilizó la función loc[] para seleccionar los registros donde la columna family_history_with_overweight es igual a "no". A continuación, para contar cada nivel de obesidad, se empleó la función value_counts(), que cuenta cuántas veces aparece cada valor único en la columna correspondiente, y se asignó este resultado a la variable conteo.

Para calcular el porcentaje, se creó otra variable llamada porcentaje_obesidad, donde cada ocurrencia de cada nivel de obesidad se divide por el número total de filas filtradas, utilizando la instrucción len(dataFiltered). Finalmente, cada valor se multiplica por 100 para obtener el porcentaje.

```
1. import pandas as pd
2. import matplotlib.pyplot as plt
3.
4. df =pd.read_csv('Obesity.csv')
5. data = pd.DataFrame(df)
6.
7. dataFiltered =
data.loc[data['family_history_with_overweight'] ==
'no']
8.
9. conteo =
dataFiltered['NObeyesdad'].value_counts()
10.
11. porcentaje_obesidad = conteo / len(dataFiltered)
* 100
12.
13. print(porcentaje_obesidad)
14.
```

Resultado:

Porcentaje de los niveles de obesidad según registros sin historial familiar de obesidad

NObeyesdad	Porcentaje sin registros de historial familiar
Insufficient_Weight	37.922078
Normal_Weight	34.285714
Overweight_Level_I	21.038961
Overweight_Level_II	4.675325
Obesity_Type_I	1.818182
Obesity_Type_II	0.259740

Tabla 5 Porcentaje de los niveles de obesidad según registros sin historial familiar de obesidad

Conclusión:

Como se puede observar en los resultados de la Tabla 5, la presencia de la obesidad en aquellas personas que no tienen familiares relacionados a esta enfermedad es muy baja en comparación con aquellos que sí tienen historial familiar de obesidad. De hecho, el mayor porcentaje proviene del nivel de 'Insufficient_Weight', al cual le sigue 'Normal_Weight'. Para finalizar, como dato a destacar, es que el nivel 'Obesity_Type_II' no presentó resultados, por lo que su porcentaje es de 0%.

Con base en la información anterior, se llegó a la conclusión de que aquellas personas que no tienen historial familiar de obesidad son menos propensas a padecer de esta. Aun así, se debe tomar en cuenta los hábitos de salud de estas personas, pues que el valor más alto haya sido peso insuficiente es preocupante. Tal vez haya una probabilidad de que no coman suficiente o ya genéticamente les cuesta subir de peso. Se insta a crear un control médico para este tipo de personas, ya que, aunque no presentan tantos padecimientos, lo ideal es poseer un peso saludable.

F. 5 personas con mayor nivel de obesidad con base en el peso

Para la solución de este problema, se utilizó la librería Pandas de Python. Primero, se importó dicha librería y se cargó el archivo CSV que contiene la información del estudio. Luego, se procedió a filtrar únicamente aquellas columnas que se desean mostrar, especificando en el código las columnas Gender, NObeyesdad y Weight. A continuación, se ordenaron los datos filtrados en orden descendente por la columna del peso utilizando el método sort_values(). Finalmente, para mostrar los primeros 5 registros, se utilizó el método head() que poseen los DataFrames.

```
1. import pandas as pd
2. import matplotlib.pyplot as plt
3.
4. df =pd.read_csv('Obesity.csv')
5. data = pd.DataFrame(df)
6.
7. dataFiltered = data[['Gender','NObeyesdad',
'Weight']]
8.
9. dataFiltered = dataFiltered.sort_values('Weight',
ascending=False)
10. print(dataFiltered.head(5))
11.
```

Resultado:

5 personas con mayor nivel de obesidad con base en el peso:

Gender	NObeyesdad	Weight
Male	Obesity_Type_III	173.00

Female	Obesity_Type_III	165.05
Female	Obesity_Type_III	160.93
Female	Obesity_Type_III	160.63
Female	Obesity_Type_III	155.87

Tabla 6 personas con mayor nivel de obesidad con base en el peso

Conclusión:

Con base en los resultados observados en la Tabla 6, se puede observar que, de los cinco resultados, las personas con más peso son mujeres en cuatro de los casos. Esto confirma los resultados anteriores: las mujeres parecen ser más propensas a alcanzar niveles extremos de obesidad (Obesidad Tipo III). Aun así, el primer resultado es de un hombre con un peso de 173.00. Por razones biológicas, se puede concluir que un hombre puede llegar a pesos más allá de lo que las mujeres pueden llegar.

Una vez más, este análisis es un llamado para la intervención a tener en cuenta sobre la necesidad de estrategias de prevención y tratamiento para las mujeres.

G. Niveles de consumo de agua según nivel de obesidad

Seguidamente, hace una estimación del promedio de consumo de agua de las personas según el nivel de obesidad. Para esto, se hace un dataframe que ordene, primeramente, los datos según los niveles de obesidad, orden el cual está establecido según el array definido como "obesity_levels" en el código abajo mostrado. Posterior a esto, se hace un promedio de la columna CH2O, la cual representa la cantidad

```
import pandas as pd
df = pd.read_csv('Obesity.csv')

obesity_levels = [
    'Insufficient_Weight',
    'Normal_Weight',
    'Overweight_Level_I',
    'Overweight_Level_II',
    'Obesity_Type_I',
    'Obesity_Type_II',
    'Obesity_Type_III'
]

df['NObeyesdad'] =
pd.Categorical(df['NObeyesdad'],

categories=obesity_levels,
                                    ordered=True)

promedio_consumo_agua =
df.groupby(['NObeyesdad'])['CH2O'].mean()

print("Promedio de consumo de agua por nivel de
obesidad:")
print(promedio_consumo_agua)

```

Resultado:

Promedio de consumo de agua según nivel de obesidad:

NObeyesdad	Promedio de consumo de agua
Insufficient_Weight	1.871281
Normal_Weight	1.850174
Overweight_Level_I	2.058725
Overweight_Level_II	2.025133
Obesity_Type_I	2.112218
Obesity_Type_II	1.877658
Obesity_Type_III	2.208493

Tabla 7 Promedio de consumo de agua según nivel de obesidad

Conclusión:

Con base en los resultados obtenidos en la tabla 7, se puede observar que los niveles de consumo de agua aumentan en una pequeña parte en las personas que presentan entre sobrepeso y obesidad tipo 1, sin embargo, en general, todos los niveles de consumo de agua son similares, con una diferencia mínima.

H. Tipos de transporte público tomados por las personas según niveles de obesidad

De una manera similar al análisis anterior, se utiliza un agrupamiento de los datos por nivel de obesidad y además por tipo de transporte público. De esta forma, se obtienen los resultados de la tabla 8. Dentro de los tipos de transporte público que toman las personas consideradas en estos datos, están automóvil, motocicleta, bicicleta, transporte público, caminar.

```
import pandas as pd
df = pd.read_csv('Obesity.csv')
obesity_levels = [
    'Insufficient_Weight',
    'Normal_Weight',
    'Overweight_Level_I',
    'Overweight_Level_II',
    'Obesity_Type_I',
    'Obesity_Type_II',
    'Obesity_Type_III'
]
df['NObeyesdad'] =
pd.Categorical(df['NObeyesdad'],

categories=obesity_levels,
                                    ordered=True)
registro_general =
df.loc[(df['MTRANS'].isin(['Automobile',

'Motorbike',

'Bike',

'Public_Transportation',

'Walking']))]

conteo_obesidad =
registro_general.groupby(['NObeyesdad',
'MTRANS']).size().unstack(fill_value=0)
print(conteo_obesidad)

```

Resultado:
Tipos de transporte que toman las personas según nivel de obesidad:

NObeyesdad	Automobile	Bike	Motorbike	Public Transportation	Walking
Insufficient_Weight	46	0	0	220	6
Normal_Weight	45	4	6	200	32
Overweight_Level_I	66	2	1	212	9
Overweight_Level_II	94	0	1	189	6
Obesity_Type_I	110	0	3	236	2
Obesity_Type_II	95	1	0	200	1
Obesity_Type_III	1	0	0	323	0

Tabla 8 Tipos de transporte que toman las personas según nivel de obesidad

Conclusión:
Con base en los resultados obtenidos en la tabla 8, es posible determinar que la mayoría de las personas consideradas en este estudio toman transporte público, sin importar su nivel de obesidad. Sin embargo, existe una pequeña diferencia en cuanto a las personas que tienen un peso normal, y es que la mayoría, después de usar transporte público, camina como forma de movilizarse. En cambio, aunque pequeña, la tendencia es que mientras mayores son los niveles de obesidad, menos tienden a caminar las personas. Con respecto a otros medios de transporte, no existe un patrón claro, ya que, por ejemplo, muy pocas personas utilizan bicicleta o motocicleta. Las personas que usan automóvil son bastantes; sin embargo, puede verse una tendencia a aumentar un poco, llegando a 110 personas con obesidad tipo 2 que conducen automóvil. En conclusión, no se pueden realizar afirmaciones muy certeras sobre estos datos, ya que la mayoría de las personas utilizan transporte público, y las que usan demás medios de transporte, aunque hay tendencias pequeñas, son una minoría.

I. Cantidad de ejercicio de las personas según nivel de obesidad

Primeramente, se realiza el orden personalizado para los niveles de obesidad, esto para tener una mejor visualización en la tabla. Seguidamente, se realiza un agrupamiento de los datos, teniendo posibles valores de 0, 1, 2 y 3 en la columna “FAF”, la cual corresponde a la cantidad de ejercicio que realizan las personas. Posterior a esto, también se ordena con base en el nivel de obesidad, para así contar los datos. Los resultados se muestran en la tabla 9, con los nombres de columnas correspondientes.

```
1. import pandas as pd
2. df = pd.read_csv('Obesity.csv')
3.
4. obesity_levels = [
5.     'Insufficient_Weight',
6.     'Normal_Weight',
7.     'Overweight_Level_I',
8.     'Overweight_Level_II',
9.     'Obesity_Type_I',
10.    'Obesity_Type_II',
11.    'Obesity_Type_III'
12. ]
13.
14. df['NObeyesdad'] =
pd.Categorical(df['NObeyesdad'],
15.
categories=obesity_levels,
16.                                    ordered=True)
17.
18. promedio_consumo_agua =
df.groupby(['NObeyesdad'])['FAF'].mean()
19.
20. print("Promedio de cantidad de ejercicio por
nivel de obesidad:")
21. print(promedio_consumo_agua)
22.
```

Resultado:

NObeyesdad	Promedio de cantidad de ejercicio a la semana
Insufficient_Weight	1.250131
Normal_Weight	1.247387
Overweight_Level_I	1.056796
Overweight_Level_II	0.958072
Obesity_Type_I	0.986748
Obesity_Type_II	0.971857
Obesity_Type_III	0.664817

Cantidad de ejercicio según nivel de obesidad:

Tabla 9 Cantidad de ejercicio según nivel de obesidad

Conclusión:
Es posible observar, con base en los resultados obtenidos en la tabla 9, que se presenta una tendencia decreciente respecto a la cantidad de ejercicio mientras mayor sea el nivel de obesidad. La cantidad más alta de ejercicio a la semana se presenta en el peso insuficiente, con un promedio de 1.25, mientras que el nivel más bajo de ejercicio se presenta en la obesidad tipo III, con 0,66.

V. Análisis de los datos

Con el fin de obtener un análisis de datos óptimo, se ha decidido optar por el uso de la librería de Matplotlib y Seaborn. Con esto, se logra una visualización más agradable y comprensible para los lectores.

A. Edad promedio según nivel de obesidad

El Gráfico 1, ofrece una visión clara de la relación entre la edad promedio y los diferentes niveles de obesidad. En él se observa que las edades promedio varían según cada categoría de obesidad.

En la categoría de peso normal, la edad promedio es alrededor de 22 años. A medida que se avanza a las categorías de sobrepeso, tanto en el nivel I como en el nivel II, la edad promedio se eleva a aproximadamente entre los 24 y 25 años. Sin embargo, en las categorías de obesidad, la edad promedio muestra variaciones interesantes. Por ejemplo, en la categoría de Obesity_Type_I, la edad promedio disminuye ligeramente a unos 25 años. En la categoría de Obesity_Type_II, la edad promedio es más alta, situándose cerca de los 28 años, y en la categoría de peso insuficiente, la edad promedio se reduce a unos 20 años.

Estas variaciones sugieren varias observaciones clave. Las personas con peso insuficiente tienden a ser más jóvenes, con edades promedio alrededor de los 20-21 años. En contraparte, las personas con sobrepeso en los niveles I y II tienen edades promedio un poco más altas. La categoría de Obesity_Type_II, con la edad promedio más alta de unos 28 años, indica que los niveles más severos de obesidad tienden a incluir personas de mayor edad.

El análisis de estos datos revela una tendencia en la cual los niveles más altos de obesidad están asociados con una edad promedio mayor. Esto podría reflejar el aumento de peso con la edad o la acumulación de problemas de salud relacionados con la obesidad a lo largo del tiempo.

```
#Gráfico de barras de edad promedio según nivel de obesidad
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('Obesity.csv')

obesity_levels = [
      'Insufficient_Weight',
      'Normal_Weight',
      'Overweight_Level_I',
      'Overweight_Level_II',
      'Obesity_Type_I',
      'Obesity_Type_II',
      'Obesity_Type_III'
]

df['NObeyesdad'] = pd.Categorical(df['NObeyesdad'],
            categories=obesity_levels,
            ordered=True)

tabla_ordenada = df.groupby(['NObeyesdad'])['Age'].mean().reset_index()

plt.figure(figsize=(9, 4))

sns.barplot(data = tabla_ordenada, x='NObeyesdad', y='Age', ci=None)

plt.title('Edad promedio según Nivel de Obesidad')

plt.xlabel('Nivel de Obesidad')
plt.ylabel('Edad Promedio')

plt.xticks(rotation=45)
plt.show()
```

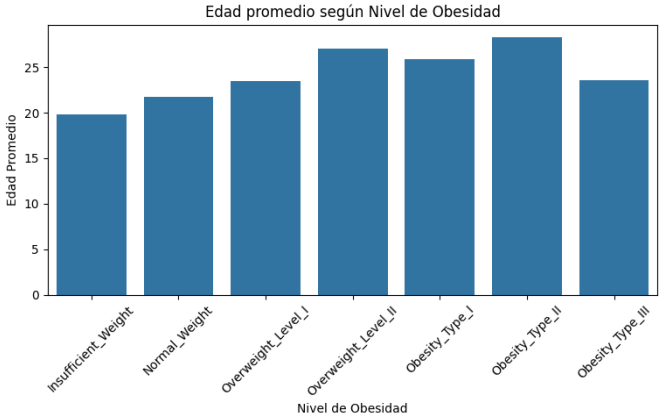

Gráfico 1 edad promedio según nivel de obesidad

B. Número de hombres y número de mujeres por nivel de obesidad

El gráfico 2, muestra la distribución del número de hombres y mujeres en distintos niveles de obesidad. En la categoría de peso normal, la cantidad de hombres y mujeres es similar, indicando un equilibrio entre ambos géneros en esta categoría.

En la categoría de Obesity_Type_I, la distribución entre hombres y mujeres es bastante equilibrada, con un número similar de individuos de ambos géneros. Sin embargo, en las categorías de obesidad más severa (Obesity_Type_II), el número de hombres es significativamente mayor que el de mujeres. Asimismo, la categoría de Obesity_Type_III muestra una diferencia notable, con una mayor cantidad de mujeres afectadas.

Por otro lado, en la categoría de peso insuficiente, se observa una mayor cantidad de mujeres en comparación con hombres, lo que sugiere que las mujeres tienen una mayor prevalencia de peso insuficiente.

Identificar estos patrones permite desarrollar estrategias de salud pública más efectivas y personalizadas, abordando adecuadamente las necesidades de diferentes grupos de género.

```
#Gráfico de barras Número de Hombres y mujeres por nivel de obesidad
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt


df = pd.read_csv('Obesity.csv')

registro_general = df.loc[(df['Gender'].isin(['Male', 'Female'])) & (df['NObeyesdad'].notnull())]

obesity_levels = [
      'Insufficient_Weight',
      'Normal_Weight',
      'Overweight_Level_I',
      'Overweight_Level_II',
      'Obesity_Type_I',
      'Obesity_Type_II',
      'Obesity_Type_III'
```

```
19. ]
20.
21. registro_general['NObeyesdad'] =
pd.Categorical(df['NObeyesdad'],
22.             categories=obesity_levels,
23.             ordered=True)
24.
25. plt.figure(figsize=(9, 4))
26.
27. sns.countplot(data=registro_general,
x='NObeyesdad', hue='Gender')
28.
29. plt.title('Número de Hombres y Mujeres por Nivel
de Obesidad')
30.
31. plt.xlabel('Nivel de Obesidad')
32.
33. plt.ylabel('Número de Personas')
34.
35. plt.xticks(rotation=45)
36.
37. plt.legend(['Femenino',
'Masculino'],title='Género')
38.
39. plt.show()
40.
```

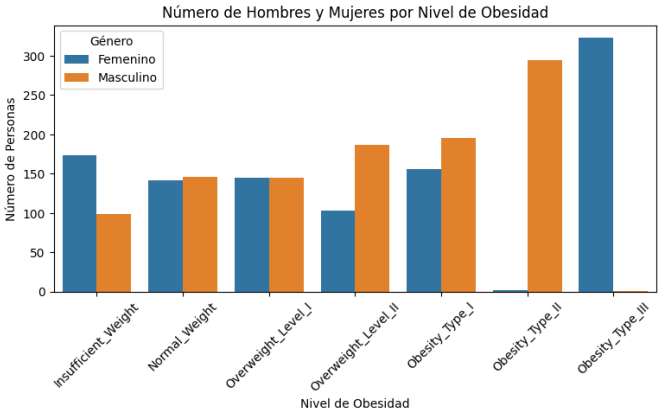

Gráfico 2 Número de hombres y número de mujeres por nivel de obesidad

C. *Porcentaje de personas según nivel de obesidad*

El gráfico 3 muestra la distribución porcentual de personas según su nivel de obesidad, ofreciendo una visión clara de cómo se dividen las distintas categorías de peso dentro de la población estudiada.

Se puede observar que las categorías de obesidad (Tipo I, II y III) en conjunto representan una parte significativa de la población, sumando aproximadamente el 46%. Esto destaca una alta prevalencia de obesidad en la muestra estudiada. Además, las categorías de sobrepeso (Nivel I y II) juntas constituyen el 27.4% de la población, indicando que más de una cuarta parte de los individuos se encuentran en un estado de sobrepeso. Por otro lado, las personas con peso normal representan una menor proporción en comparación con aquellas que tienen algún nivel de obesidad o sobrepeso, lo cual puede sugerir una tendencia hacia un mayor índice de masa corporal en la población estudiada. Aunque menos prevalente, el peso insuficiente afecta al 12.9% de los individuos, lo cual sigue siendo una preocupación significativa.

La alta prevalencia de obesidad y sobrepeso sugiere la necesidad de intervenciones específicas para prevenir y reducir la obesidad, así como para promover hábitos alimenticios saludables y la actividad física. Además, la considerable proporción de personas con peso insuficiente resalta la importancia de implementar estrategias para combatir la malnutrición.

```
 1. import pandas as pd
 2. import seaborn as sns
 3. import matplotlib.pyplot as plt
 4.
 5. # Cargar los datos desde el archivo CSV
 6. df = pd.read_csv('Obesity.csv')
 7.
 8. # Renombrar las etiquetas de los niveles de
obesidad
 9. df['NObeyesdad'] = df['NObeyesdad'].replace({
10.     'Obesity_Type_I': 'Obesidad Tipo I',
11.     'Obesity_Type_III': 'Obesidad Tipo III',
12.     'Obesity_Type_II': 'Obesidad Tipo II',
13.     'Overweight_Level_II': 'Sobrepeso Nivel II',
14.     'Overweight_Level_I': 'Sobrepeso Nivel I',
15.     'Normal_Weight': 'Peso Normal',
16.     'Insufficient_Weight': 'Peso Insuficiente'
17. })
18.
19. # Contar el número de personas en cada nivel de
obesidad
20. cant_obesidad = df['NObeyesdad'].value_counts()
21.
22. print(cant_obesidad)
23. # Configurar el tamaño del gráfico
24. plt.figure(figsize=(8, 8))
25.
26. # Crear el gráfico de pastel
27. plt.pie(cant_obesidad,
labels=cant_obesidad.index, autopct='%1.1f%%',
startangle=140, colors=sns.color_palette('tab10',
len(cant_obesidad)))
28. plt.title('Porcentaje de personas según nivel de
obesidad')
29. plt.axis('equal')
30.
31. # Mostrar el gráfico
32. plt.show()
33.
```

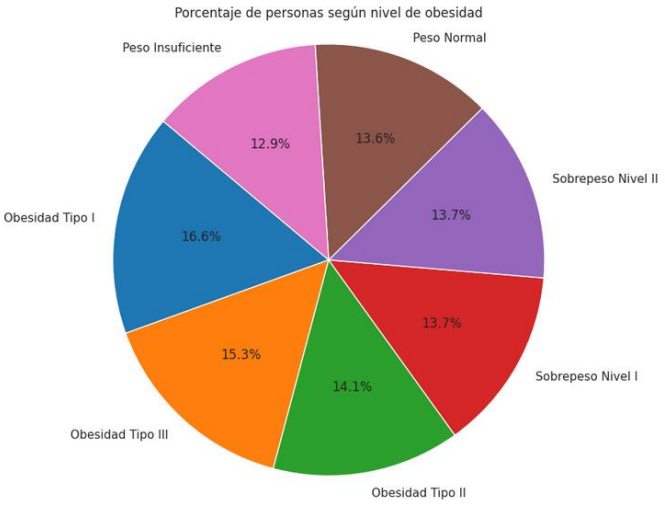

Gráfico 3 Porcentaje de personas según nivel de obesidad

VI. CONCLUSIONES

Con el presente artículo, al realizar un análisis exhaustivo de los datos, se observó que ciertos hábitos alimenticios o estilo de vida, como la falta de ejercicio, así como la insuficiente ingesta de agua, tienden a contribuir al desarrollo de la obesidad. Todos estos aspectos recalcan la importancia de considerar el estilo de vida al abordar este problema. Con base en los resultados, diseñar estrategias de prevención y tratamiento más efectivas contra esta enfermedad se vuelve crucial.

Tampoco se puede dejar de lado que, aunque los hábitos mencionados son controlables, otros no lo son tanto, como el género y el historial familiar de obesidad. En el caso del género, se observa que las mujeres son particularmente vulnerables a sufrir obesidad tipo III, mientras que los hombres tienden a padecer obesidad tipo II. Por otro lado, tener un historial familiar de obesidad también influye significativamente; de hecho, se llegó a la conclusión de que aquellas personas que no poseen historial familiar de obesidad son menos propensas a padecerla.

Otro punto importante a destacar como resultado de la investigación es que la mayoría de las personas que funcionaron como fuentes de información para los investigadores presentan algún tipo de obesidad. Es preocupante que un valor tan alto como el 73.4% de los participantes se encuentre en esta condición. Se resalta la urgencia de abordar la obesidad como un problema de salud pública.

Asimismo, la elección de Python como lenguaje de programación es importante, ya que es una de las herramientas más populares para el análisis de datos. Ofrece varias ventajas para aquellos que no tienen experiencia en programación: facilidad de uso y lectura, bibliotecas especializadas como Pandas y Matplotlib, creación de gráficos y utilización de funciones predefinidas. El uso de este lenguaje ayuda a interpretar y comprender una gran cantidad de información.

Como parte final de este artículo, se recalca que la ciencia abierta ha sido extremadamente importante durante la investigación, ya que ha ayudado a obtener la información para realizar un análisis de obesidad con datos reales. Además, ofrece un acceso igualitario a cualquier persona, independientemente de los recursos económicos o la ubicación geográfica, considerando que, en su mayoría, los datos que las entidades poseen son de gran confiabilidad.

VII. RECOMENDACIONES

Dado el anterior análisis de los datos sobre la obesidad en función de ciertos aspectos, tales como la ingesta de agua, actividad física, etc. es posible resaltar recomendaciones asociadas a los resultados de este estudio.

Primeramente, la actividad física es una de las actividades más importantes para tener en cuenta para evitar la obesidad. En el análisis previo visto en el **punto I,** la cantidad de actividad física que hacían las personas iba disminuyendo conforme los niveles de obesidad eran más altos, por lo que se insta a las personas a realizar ejercicio de una manera periódica.

Como segundo punto, es importante determinar diferentes estrategias dependiendo del género, ya que, según el análisis antes visto en el punto A, existe una mayor cantidad de mujeres que presentan un nivel de obesidad alto. Por ello, es necesario determinar diferentes estrategias dependiendo del género, preferiblemente con un experto en el tema.

Como tercer punto, las personas con familia con historial de obesidad necesitan establecer planes de prevención contra esta enfermedad, ya que se ha visto que este es un factor que está relacionado con los niveles altos de obesidad.

Como última recomendación, es importante que, aun sin tener ningún factor de riesgo asociado a esta enfermedad, se tomen medidas que ayuden a tener una vida saludable, como llevar controles de dieta. Actualmente, la tecnología permite explorar diferentes medios con los cuales se puede tener un seguimiento de, por ejemplo, la cantidad de ejercicio que se hace al día. El primer paso para prevenir es concientizarse sobre la existencia de este problema, y poseer una iniciativa para poder llevar a cabo estas estrategias.

VIII. REFERENCIAS

[1] X. Liu and H. Xu, "School-Enterprise Cooperation on Python Data Analysis Teaching," in *2019 14th International Conference on Computer Science & Education (ICCSE)*, Toronto, ON, Canada: IEEE, 2019, pp. 278–281. doi: 10.1109/ICCSE.2019.8845524.

[2] A. Nagpal and G. Gabrani, "Python for Data Analytics, Scientific and Technical Applications," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, Dubai, United Arab Emirates: IEEE, 2019, pp. 140–145. doi: 10.1109/AICAI.2019.8701341.

[3] J. Singh, J. Singh, G. Singh, and N. Kaur, "Exploratory Data Analysis for Interpreting Model Prediction using Python," in *2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, Bangalore, India: IEEE, Dec. 2022, pp. 1–6. doi: 10.1109/SMARTGENCON56628.2022.10083533.

[4] T. G. Mattson, T. A. Anderson, and G. Georgakoudis, "PyOMP: Multithreaded Parallel Programming in Python," Comput. Sci. Eng., vol. 23, no. 6, pp. 77–80, Nov. 2021, doi: 10.1109/MCSE.2021.3128806.

[5] P. Bhardwaj, C. Choudhury, and P. Batra, "Automating Data Analysis with Python: A Comparative Study of Popular Libraries and their Application," in *2023 3rd International Conference on Technological*

Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan: IEEE, Nov. 2023, pp. 1243–1248. doi: 10.1109/ICTACS59847.2023.10390032.

[6] G. Yu, "Financial data analysis and risk quantification based on Python," in *2021 International Conference on Computer, Blockchain and Financial Development (CBFD)*, Nanjing, China: IEEE, 2021, pp. 214–217. doi: 10.1109/CBFD52659.2021.00049.

[7] L. Li, "Big Data Analysis of Video Index Popularity and Public Emotion Based on Python," in *2023 IEEE International Conference on Electrical, Automation and Computer Engineering (ICEACE)*, Changchun, China: IEEE, Dec. 2023, pp. 375–379. doi: 10.1109/ICEACE60673.2023.10442457.

[8] A. Abodayeh, R. Hejazi, W. Najjar, L. Shihadeh, and R. Latif, "Web Scraping for Data Analytics: A BeautifulSoup Implementation," in *2023 Sixth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU)*, Riyadh, Saudi Arabia: IEEE, 2023, pp. 65–69. doi: 10.1109/WiDS-PSU57071.2023.00025.

[9] I. Grout, W. A. P. De Ferreira, and A. C. R. D. Silva, "On-Line Electrical Supply Generation Fuel Mix Data Analysis using Python and TensorFlow," in *2019 International Conference on Power, Energy and Innovations (ICPEI)*, Pattaya, Chonburi, Thailand: IEEE, 2019, pp. 20–23. doi: 10.1109/ICPEI47862.2019.8944972.

[10] Y. Wang, L. Xu, Q. Wang, H. Lv, and Y. Zhang, "Educational Data Mining and Learning Analysis System Based on Python," in *2022 12th International Conference on Information Technology in Medicine and Education (ITME)*, Xiamen, China: IEEE, 2022, pp. 559–563. doi: 10.1109/ITME56794.2022.001