TRABAJO DE GRADO
Opción Seminario-Diplomado.

ALGORITMO COMPUTACIONAL PARA EL ANÁLISIS Y TOMA DE DECISIONES EN DATOS DE RIESGO DE OBESIDAD, UTILIZANDO ESTRATEGIAS DE MACHINE LEARNING

Corporación Universitaria Remington.
Facultad de ingeniería de sistemas
Seminario de machine learning en tiempo de datos

Estudiante:
Dahiana Calderon Gonzalez
Tutor: Juan Carlos Briñez de León
Opción de Trabajo de grado Seminario-Diplomado.
2024.

Tabla de Contenidos

Contenido

Resumen

Según la OMS la tasa de niveles de obesidad entre jóvenes y adultos se ha ido incrementando con el paso del tiempo. por esto, se ha dado la tarea de la implementación de un algoritmo de machine Lear Ning de aprendizaje supervisado enfocado en el ámbito de la salud de los usuarios, este algoritmo se centra en desarrollar métodos que agilicen los diagnósticos médicos en pacientes que lo requieran, sin la necesidad de presentarse en un centro médico. Este sistema cuenta con la facilidad de entender los valores que se le brinden, buscando así predecir si un usuario cuenta con obesidad según su peso, altura, y su manejo de hábitos diarios. Estos datos obtenidos por los individuos serán tomados en cuenta para identificar en que nivel de obesidad se encuentra.

Este algoritmo implementado en este trabajo, cuenta con la información de 20758 datos de personas jóvenes, adultos- jóvenes y adultos con la finalidad de dar a conocer que la obesidad es una enfermedad crónica que se debe tener en cuenta y no dejarla pasar por desapercibido.

Palabras clave

Machine Lear Ning, clasificación de datos, riesgos de obesidad, predicciones, Salud.

Marco conceptual y contextual

El presente texto corresponde a los conceptos primordiales que se presentarán en el documento, como también del método que utilizaremos para realizar el análisis de estos datos, esto con el fin de contextualizar los temas abarcados llevando un desarrollo y entendimiento efectivo del trabajo.

Obesidad.

Según las referencias encontradas la obesidad es una enfermedad crónica que se está presentando en todo el mundo. Se caracteriza principalmente por la acumulación de grasa en exceso en diferentes partes del cuerpo humano. Dependiendo de su aumento en algunas zonas corporales, se pueden identificar riesgos en la salud que pueden llegar a ser un factor importante de mortalidad en el mundo [1].

Debido a esto, la obesidad ha sido clasificada, (según la OMS) en una tabla, la cual se ha visto en el documento "Causas de obesidad" [1] de la siguiente manera:

Clasificación	**IMC (kg/m2)**	**Riesgo Asociado a la salud**
normo peso	18.5-24.9	Promedio
exceso de peso	≥ 25	
Sobrepeso o Pre-Obeso	25 - 29.9	AUMENTADO

Obesidad Grado I o moderada	30 − 34.9	AUMENTADO MODERADO
Obesidad Grado II o severa	35 - 39.9	AUMENTO SEVERO
Obesidad Grado III o mórbida	≥ 40	AUMENTO MUY SEVERO

Tabla1: clasificación de obesidad según la OMS.

Alimentación saludable.

Según lo investigado, La alimentación saludable se define como el indicativo primordial para un funcionamiento corporal optimo del cuerpo. en el cual, también se obtienen diferentes beneficios. tales como, la reducción de riesgos de enfermedades, mantener, regular y mejorar la salud, como también impulsar el crecimiento y desarrollo en niños y jóvenes.[2]

Actividad física.

Según las referencias que se encontraron, La actividad física se puede clasificar en dos conceptos, esto con el fin de diferenciar dos términos que se encuentran presentes ante esta definición pero que se encuentran actualmente en la sociedad definidas de manera conjunta. En primera parte se encuentra la actividad física, se caracteriza por ser las actividades que realiza una persona en su día a día de manera continua, la cual hace un consumo de energía debido a los movimientos que se realizan con el cuerpo. Por otro lado, encontramos el ejercicio físico, que es la actividad que realizamos de una forma intencionada, en algunos casos de manera continua y diaria, su objetivo es mantener un estado físico corporal optimo. [3]

Google Colaboratory.

Google Colaboratory es una herramienta de solución en el ámbito de aprendizaje automático, debido a que es un servicio alojado de Jupyter Notebook [4]y sus entornos se encuentran preconfigurados en lenguaje Python. Esto con la finalidad de evitar instalar lenguajes en el ordenador. gracias a su acceso gratuito y su accesible forma de utilizarlo desde una cuenta de Google, hoy en día Colab ha llegado a ofrecer un amplio desarrollo efectivo en el campo de la docencia. [4,5]

Aprendizaje supervisado

el aprendizaje supervisado consta de un tipo de aprendizaje automático donde a este se le brinda información sobre los datos que se le están presentando, es decir, cada dato contiene su propia etiqueta que hace que este lo defina y pueda ser clasificado con etiquetas iguales. El objetivo principal de este aprendizaje supervisado, en primer lugar, que se analicen los datos que se le están brindando, también llamados como datos de entrada. los cuales en segundo lugar hará automáticamente su clasificación debido a las características que contengan, para finalmente realizar una salida de variable precisa de estos datos.

En cuanto a este aprendizaje supervisado el cual se estará desarrollando este documento, la (figura 1) presentará la clasificación de como se le enseñará al clasificador los datos.[6]

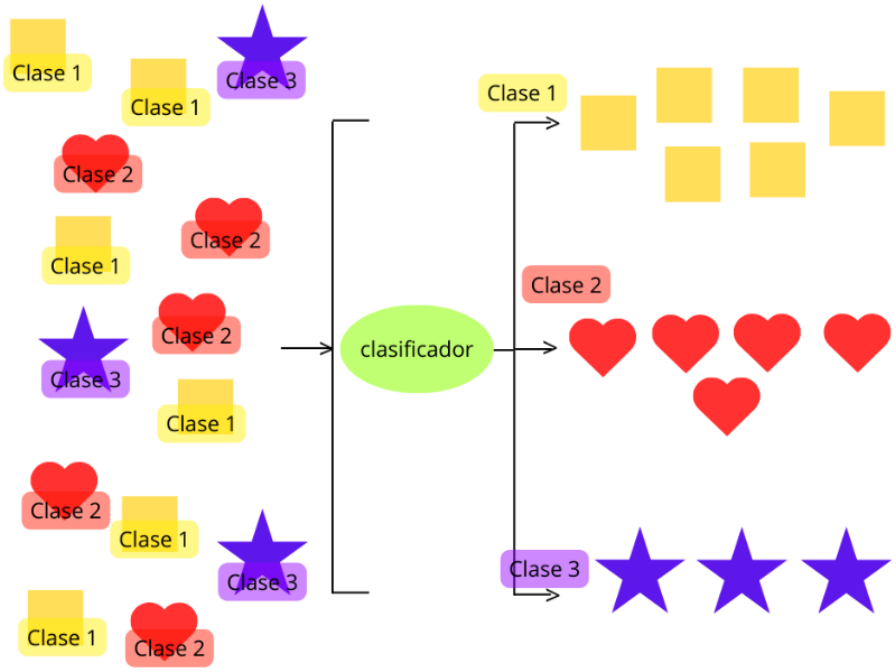


Figura 1. Aprendizaje supervisado. Fuente: Elaboración propia

El aprendizaje supervisado contiene un lado importante, del cual en el desarrollo del presente documento será el aprendizaje que utilizaremos. Este consiste en el método de clasificación, en el cual los datos que ya se le ensañaron al sistema, este al obtener un nuevo dato, obtendrá su respuesta inmediatamente debido a su aprendizaje explicado en la (figura 1) para esto la (figura 2) se observa como el sistema responderá al momento de recibir un nuevo dato.[6]

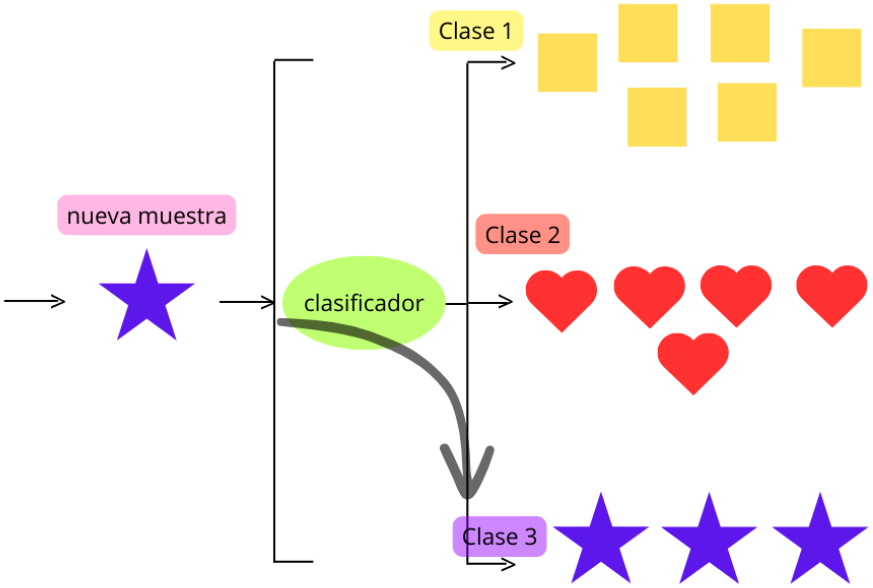


Figura 2. Aprendizaje supervisado, clasificación. Fuente: Elaboración propia.

Pregunta problema

¿Cómo implementar un modelo de machine learning que desarrolle y optimice una base de datos, consiguiendo la clasificación de obesidad en una persona?

Acercamiento a los datos:

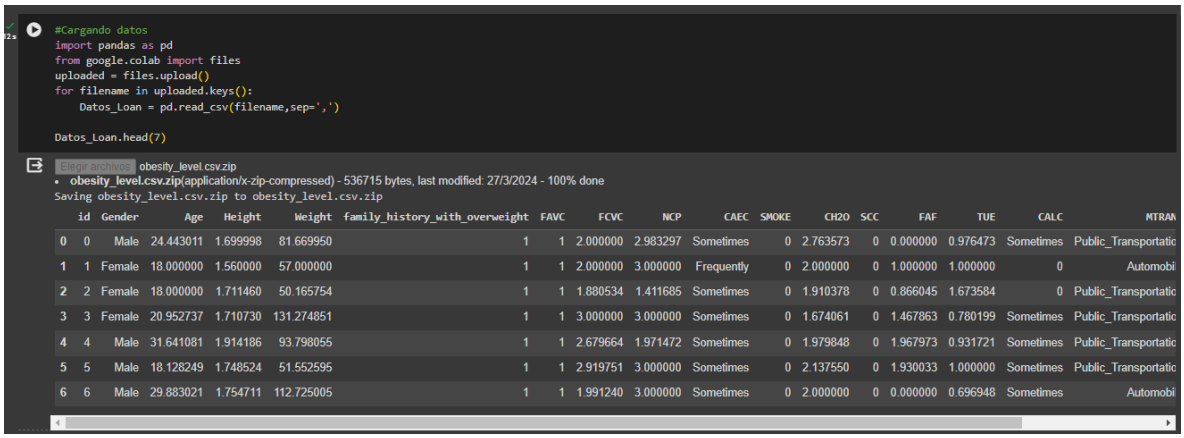
Este conjunto de datos llamado “Obesity Risk Dataset” (Conjunto de datos sobre riesgo de obesidad) presentada por “Jayesh Jain (Owner)” [7]. Se ha elegido desde la plataforma kaggle, Esta es una página que nos brinda información completa de temas específicos en datos, ya sea en formato cvs.txt. esta página logra obtener buena vista para las personas interesadas en los análisis de datos debido a su facilidad de tenerlos sin ninguna limitación

de obtención. En este caso, el conjunto de datos que se ha elegido, se presentan la información de las personas que se encuentran en este conjunto de datos del riesgo de obesidad.[7]

Descripción de variables.

Este conjunto de datos se encuentra conformado por 18 etiquetas las cuales se clasificaron en tres conjuntos de datos. Los datos que contienen datos números enteros, los datos que contienen números decimales y los datos de tipo carácter. [7] (figura 3)

Para obtener una visión más precisa sobre los datos, y verificar correctamente en que tipo se encuentra, se importan los datos y se hecha un vistazo sobre ellos en la plataforma Colab. (figura 3)(figura 4)



```
#Cargando datos
import pandas as pd
from google.colab import files
uploaded = files.upload()
for filename in uploaded.keys():
    Datos_Loan = pd.read_csv(filename,sep=',')

Datos_Loan.head(7)
```

Elegir archivos obesity_level.csv.zip

- **obesity_level.csv.zip**(application/x-zip-compressed) - 536715 bytes, last modified: 27/3/2024 - 100% done

Saving obesity_level.csv.zip to obesity_level.csv.zip

	id	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRAN
0	0	Male	24.443011	1.699998	81.669950	1	1	2.000000	2.983297	Sometimes	0	2.763573	0	0.000000	0.976473	Sometimes	Public_Transportatic
1	1	Female	18.000000	1.560000	57.000000	1	1	2.000000	3.000000	Frequently	0	2.000000	0	1.000000	1.000000	0	Automobi
2	2	Female	18.000000	1.711460	50.165754	1	1	1.880534	1.411685	Sometimes	0	1.910378	0	0.866045	1.673584	0	Public_Transportatic
3	3	Female	20.952737	1.710730	131.274851	1	1	3.000000	3.000000	Sometimes	0	1.674061	0	1.467863	0.780199	Sometimes	Public_Transportatic
4	4	Male	31.641081	1.914186	93.798055	1	1	2.679664	1.971472	Sometimes	0	1.979848	0	1.967973	0.931721	Sometimes	Public_Transportatic
5	5	Male	18.128249	1.748524	51.552595	1	1	2.919751	3.000000	Sometimes	0	2.137550	0	1.930033	1.000000	Sometimes	Public_Transportatic
6	6	Male	29.883021	1.754711	112.725005	1	1	1.991240	3.000000	Sometimes	0	2.000000	0	0.000000	0.696948	Sometimes	Automobi

Figura 3. Código de Importación de los datos en Colab. Fuente: Elaboración propia

```
Datos_Loan.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20758 entries, 0 to 20757
Data columns (total 18 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              20758 non-null  int64
 1   Gender                          20758 non-null  object
 2   Age                             20758 non-null  float64
 3   Height                          20758 non-null  float64
 4   Weight                          20758 non-null  float64
 5   family_history_with_overweight  20758 non-null  int64
 6   FAVC                            20758 non-null  int64
 7   FCVC                            20758 non-null  float64
 8   NCP                             20758 non-null  float64
 9   CAEC                            20758 non-null  object
 10  SMOKE                           20758 non-null  int64
 11  CH2O                            20758 non-null  float64
 12  SCC                             20758 non-null  int64
 13  FAF                             20758 non-null  float64
 14  TUE                             20758 non-null  float64
 15  CALC                            20758 non-null  object
 16  MTRANS                          20758 non-null  object
 17  0be1dad                         20758 non-null  object
dtypes: float64(8), int64(5), object(5)
memory usage: 2.9+ MB
```

figura 4. Código de visualización del tipo de dato al que corresponden las etiquetas. Fuente: Elaboración propia

teniendo en cuentas la información dada en la (figura 4) se determina la clasificación de 3 tipos de conjunto de datos.

En primer lugar, se encuentran los datos que contienen datos números enteros los cuales son:

1. Id
2. Historia familiar con sobrepeso
3. FAVC (Consumo frecuente de alimentos ricos en calorías)
4. SMOKE (si fuma o no)
5. SCC (Consumo de bebidas calóricas)

En segundo lugar, se encuentran los datos que contiene números decimales los cuales son:

1. Edad
2. Altura
3. Peso
4. FCVC (Frecuencia de consumo de vegetales)
5. NCP (Número de comidas principales)
6. CH2O (Consumo diario de agua)
7. FAF (Frecuencia de actividad física)
8. TUE (Tiempo de uso de dispositivos tecnológicos)

Por último, se encuentran los datos que contienen datos tipos carácter los cuales son:

1. Genero
2. CAEC (Consumo de alimentos entre comidas)
3. CALC (Consumo de alcohol)
4. MTRANS (Modo de transporte)
5. 0be1dad (Variable objetivo que representa el nivel de obesidad)

Posibles aplicaciones.

En cuento a el tema en el que estamos basando este análisis sobre datos de riesgos de obesidad en personas jóvenes y adultos-jóvenes. Se lleva a una clasificación reducida de elecciones a los cuales aplicaría en la sociedad. Por esta razón, será solamente clara en el campo de la salud. Por un parte, al tener el acceso a ello, usuarios que tengan dudas sobre

su estado físico, puedan identificar en que clasificación de obesidad se encuentre sin la necesidad de presentarse a un centro médico. Esto con el fin de simplemente informar y sugerir que debe presentarse a su médico si su estado físico se encuentra en riesgo. Por otro lado, se puede impactar en el campo de la salud de una manera técnica, ayudándole a médicos y especialistas. Implementando modelos de machine learning que ayuden a identificar el estado físico de una persona al momento de que realicen un chequeo médico agilizando los tiempos y una respuesta rápida para los pacientes.

Aproximaciones con gráficos.

Teniendo en cuenta la variedad de informacion que se encuentran en la base de datos correspondiendo a los riesgos de obesidad, se genera un grafico para los valores considerados mas importantes. esto con el fin de describir brevemente que tipos de datos se analisaran y por ende mostrar la importancia de ellos.

- En primera parte este codigo (figura 4), mostrará los niveles de obesidad que se encuentran en la columna “0be1dad”, este se interpreta como la base importante del mismo debido a que es el tema principal del dataframe.

```
#Cargando librerías
import seaborn as sns
import matplotlib.pyplot as plt

# Crear un gráfico de torta
frecuencias = Datos_Loan['0be1dad'].value_counts()
plt.figure(figsize=(8, 8))  # Tamaño del gráfico (opcional)
plt.pie(frecuencias, labels=frecuencias.index, autopct='%1.1f%%', startangle=140)
plt.title('Diagrama de Torta de Distribución de clasificacion de obesidad')
plt.show()
```

Figura 5. Código de diagrama de torta, clasificación de obesidad. Fuente: Elaboración propia.

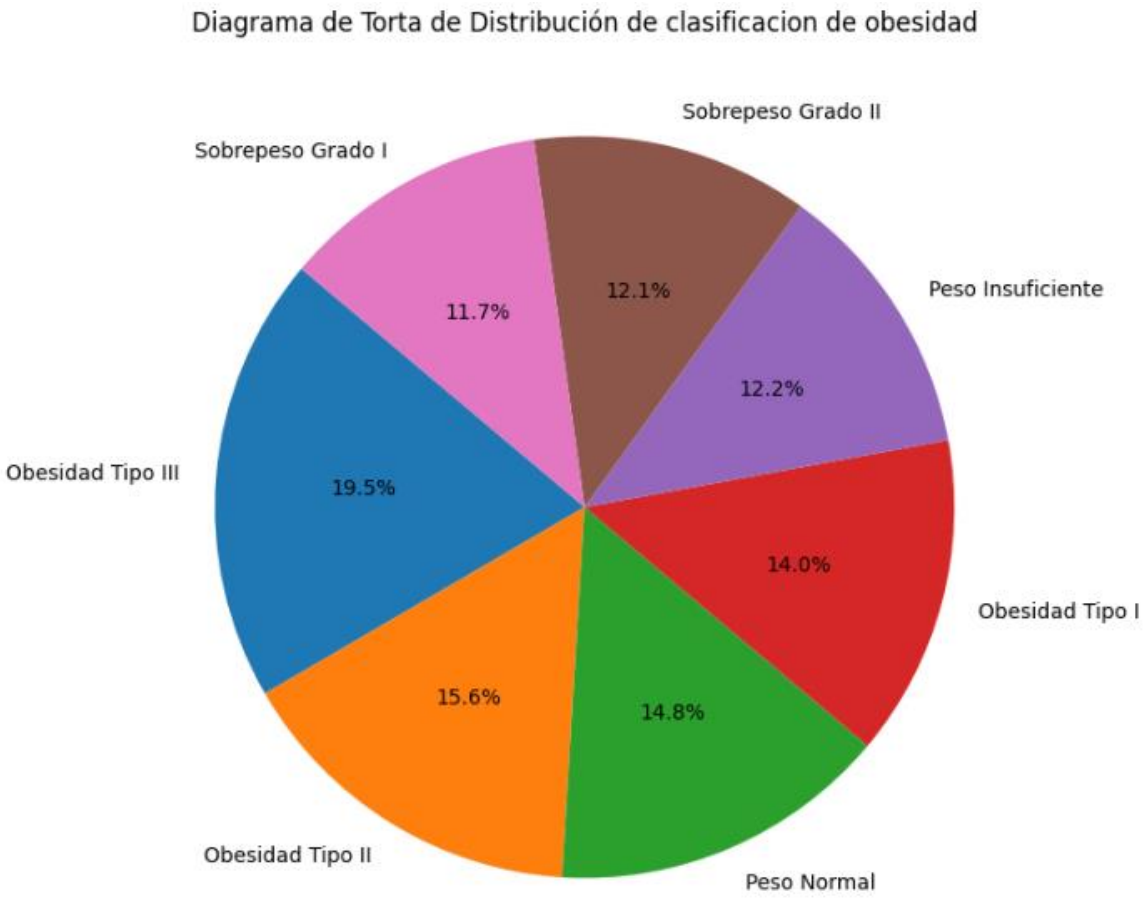


Figura 6. Diagrama de torta de distribución de clasificación de obesidad. Fuente: Elaboración propia.

- En vista del resultado (figura 5), el 19.5% de las personas registradas en la base de datos, tiene obesidad tipo III siendo el mayor valor de datos en comparación a las otras clasificaciones. Sin embargo, las clasificaciones siguientes a estas, como lo son: Obesidad tipo I, Obesidad tipo II y Peso normal contienen una disparidad breve. De la misma forma se encuentra la clasificación de Peso insuficiente, sobrepeso grado I y sobre peso grado II.

- Como siguiente, la (figura 7) corresponde al código en donde se han importado los datos de la columna “Gender” del dataframe, en el cual los datos se les permite mostrar la cantidad del valor por género sobre X y por ende la frecuencia en Y.

```
#Cargando librerías
import seaborn as sns
import matplotlib.pyplot as plt

# Crear un gráfico de barras con Seaborn
sns.countplot(data=Datos_Loan, x='Gender')
plt.title('Distribución del género')
plt.xlabel('Genero')
plt.ylabel('Frecuencia')
plt.show()
```

Figura 7. Código de grafico de barras de distribución de género. Fuente: Elaboración propia.

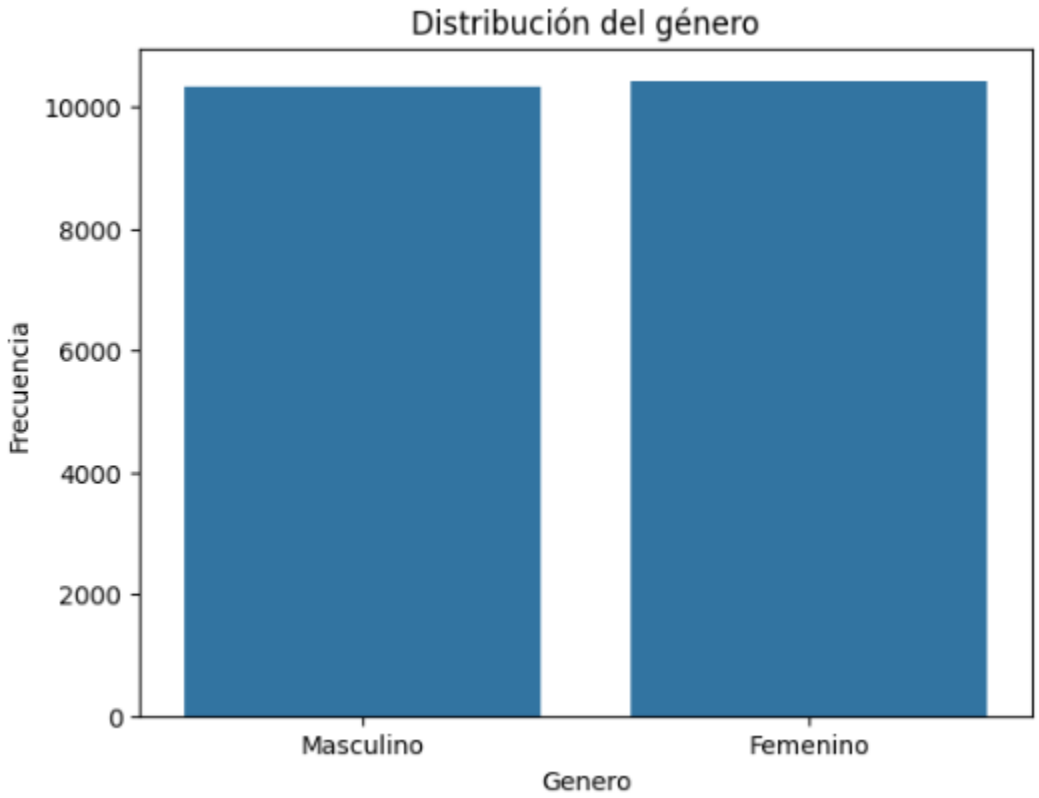


Figura 8. Grafico de barras de la distribución del género. Fuente: Elaboración propia.

- Al igual que la (figura 8) se ha tenido en cuenta los datos que la conforman para implementar la (figura 9) dado que los datos anteriores tienen una disparidad mínima la cual visualmente no es clara. por tanto, en la siguiente figura (figura 9) notamos los porcentajes de manera detallada en un grafico de torta de dispersión, Determinando sus áreas y sus diferencias porcentual, como se puede evidenciar a continuación.

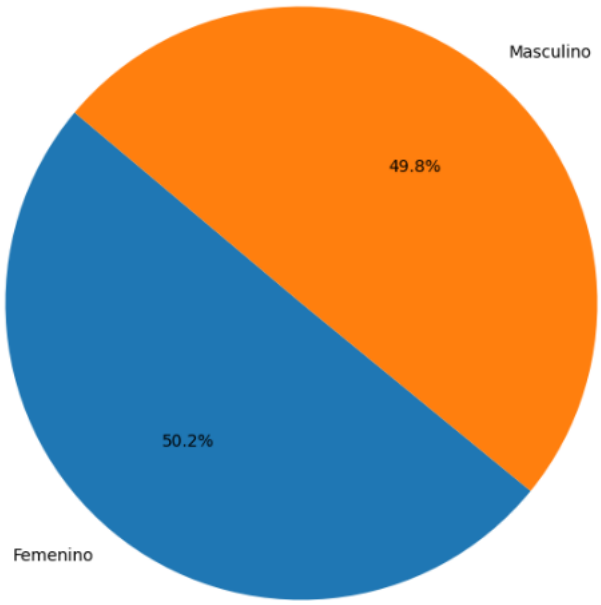


Figura 9. Diagrama de torta de distribución de clasificación de género.

- Se puede observar en este diagrama de torta que a pesar de su desigualdad mínima entre las cantidades de los datos de género, el mayor índice se encuentra en personas de sexo femenino con un valore del 50.2% en comparación de sexo masculino con un valore del 49.8%.

- Por otro lado, se presentarán figuras de las cuales obtendrán información relevante de la posible causa más importante que se tienen en común de las personas con obesidad, sobrepeso o peso normal.
- Una de las causas del sobrepeso en las personas, es la falta de actividad física, en este caso (Figura 10, 11) se realizo un mapa de calor el cual tendrá como solución los diferentes medios de transporte de las personas que conforman este dataframe. Se importa la información (figura 10) de las columnas “0b1edad” la cual consiste en la clasificación de obesidad y “MTRANS” el cual consiste en el medio de transporte utilizado por las personas que conforman este dataframe.

```
[14] from matplotlib import pyplot as plt
     import seaborn as sns
     import pandas as pd
     plt.subplots(figsize=(8, 8))
     df_2dhist = pd.DataFrame({
         x_label: grp['0be1dad'].value_counts()
         for x_label, grp in _df_23.groupby('MTRANS')
     })
     sns.heatmap(df_2dhist, cmap='viridis')
     plt.xlabel('MTRANS')
     _ = plt.ylabel('0be1dad')
```

Figura 10. Código de mapa de calor de obesidad por medio de transporte. Fuente: Elaboración propia.

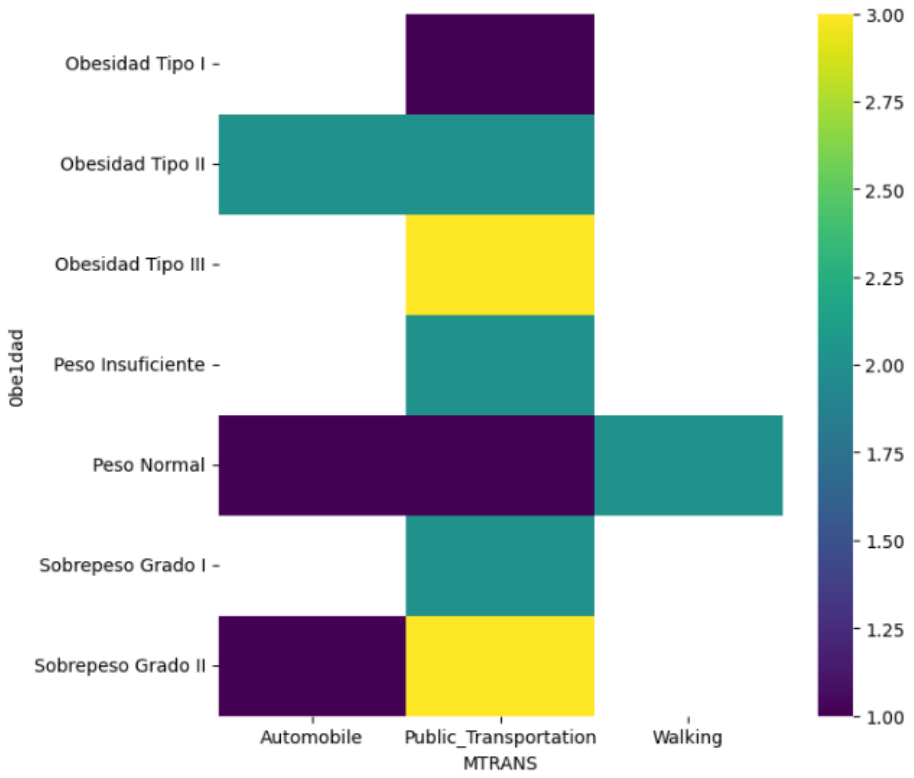


Figura 11. Mapa de calor, obesidad por medio de transporte. Fuente: Elaboración propia.

- Según lo que podemos notar (figura 11) los colores son la forma en la que se presenta la cantidad de usuarios que estan en esa escala representandose por una variasion de colores. En este caso de tonos violeta a las clasificaciones que son menos seleccionadas y los tonos amarillos a las clasificaciones más seleccionadas, por ende, en primer lugar, se puede interpretar que las personas de todas las clasificaciones de obesidad su medio de tranporte mas utilizado es el público. En segundo lugar, se observa como los usuarios que presentan sobrepeso grado II y obesidad tipo II son las personas que utilizan frecuentemente este medio de transporte.

Objetivos:

Objetivo general.

Implementar un algoritmo computacional de aprendizaje supervisado (clasificación) de machine learning. Utilizando datos de riesgos de obesidad para identificar patrones, obtener análisis y generar toma de decisiones.

Objetivos específicos.

- Determinar y ejecutar la base de datos para conocer su información, y clasificar los datos más relevantes.
- Diseñar e Implementar la arquitectura de un algoritmo de aprendizaje supervisado, Junto con la información brindada del dataframe, para entrenar el algoritmo de machine learning.
- Verificar el desempeño del algoritmo de aprendizaje supervisado, para la obtención y precisión de la toma de decisiones.
- Validar el funcionamiento del algoritmo de toma de decisiones a partir de datos nuevos.

Desarrollo e implementación del aprendizaje

El proyecto desarrollado está dentro de un entorno el cual consiste en la implementación de un algoritmo de machine learning que se ha extendido en la rama del aprendizaje supervisado por el método de clasificación. Partiendo de esto, se implemento el datframe que contiene datos sobre el riesgo de obesidad, de forma que este brinde su información al algoritmo y pueda aprender de ellos y conseguir predicciones o en este caso generar diagnósticos. Para obtener mejores resultados se conto con una serie de modelos

los cuales son previamente entrenados para alcanzar predicciones concretas. Fueron seleccionados 4 modelos los cuales son: en primer lugar, esta KNN (K-Nearest Neighbor) el cual es un clasificador utilizado muy a menudo en ejercicios de clasificación, este proporciona predicciones rápidas. su técnica de predicción basa en tener en cuenta la información que tiene sus vecinos, es decir, su método es comparar el dato nuevo por datos que ya han sido entrenados y el cual tengan una similitud para así mismo dar su resultando [8]. En segundo lugar, esta BAYES, el cual es un método de aprendizaje que consiste en calcular la probabilidad y las similitudes de un nuevo dato en las clases ya entrenadas y elige su decisión dependiendo de sus características o similitudes más acertadas [9]. En tercer lugar, esta LDA (Linear Discriminant Analysis), se basa en encontrar las dimensiones a la cuales se está presentando los datos haciendo una combinación lineal de las características que asemejen y categorizándolos. De esta forma al ingresar un nuevo dato, este se optimice entre grupos eligiendo el más acertado [10]. Por último, esta SVM (support Vector Machine) según referencias encontradas [11] consiste en dividirse los datos en dos grupos por medio de una función lineal. Al ser usado para predecir, su propósito es descomponer en la línea que divide los dos grupos buscando sus mejores características para predicciones concretas [11]. Gracias a estos modelos finalmente se harán una serie de preguntas, las cuales son las principales causas que se registraron en el dataframe y que los usuarios que brindaron esta información tienen en común. por lo tanto, estas conformaran las preguntas las cuales el usuario tendrá que ingresar al algoritmo para así obtener finalmente su diagnóstico.

Procesamiento de los datos

Importar datos 1.1.

Como primer paso, se importan los datos para visualizarlos en el algoritmo. se carga el archivo tipo csv ya antes descargado y guardado en el equipo. Ya cargados los datos la información se mostrara en una tabla por columnas. (figura 12) En el cual se verán las variables que la conforman, en este caso van a estar todas las probables causas de obesidad, genero, edad, y niveles de obesidad.

```
#Cargando datos
import pandas as pd
from google.colab import files
uploaded = files.upload()
for filename in uploaded.keys():
    Datos_Loan = pd.read_csv(filename,sep=',')

Datos_Loan.head(7)
```

Elegir archivos obesity_level.csv.zip

- **obesity_level.csv.zip**(application/x-zip-compressed) - 536715 bytes, last modified: 27/3/2024 - 100% done

```
Saving obesity_level.csv.zip to obesity_level.csv.zip
```

	id	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	
0	0	Male	24.443011	1.699998	81.669950	1	1	2.000000	2.983297	Sometimes	0	2.76
1	1	Female	18.000000	1.560000	57.000000	1	1	2.000000	3.000000	Frequently	0	2.00
2	2	Female	18.000000	1.711460	50.165754	1	1	1.880534	1.411685	Sometimes	0	1.91
3	3	Female	20.952737	1.710730	131.274851	1	1	3.000000	3.000000	Sometimes	0	1.67
4	4	Male	31.641081	1.914186	93.798055	1	1	2.679664	1.971472	Sometimes	0	1.97
5	5	Male	18.128249	1.748524	51.552595	1	1	2.919751	3.000000	Sometimes	0	2.13
6	6	Male	29.883021	1.754711	112.725005	1	1	1.991240	3.000000	Sometimes	0	2.00

figura 12. Importación de los datos. Fuente: Elaboración propia.

Conocer los Datos 2.1

Como segundo paso, se usa la función “Datos_Loan.info()” (figura 13) que nos dará desde una perspectiva más clara los datos que se estarán manejando, para obtener

primeramente la cantidad de los datos y características más precisas de ellos. además de verificar correctamente en que tipos se encuentran. Es decir, si son de tipo caracter, tipo float o tipo int.

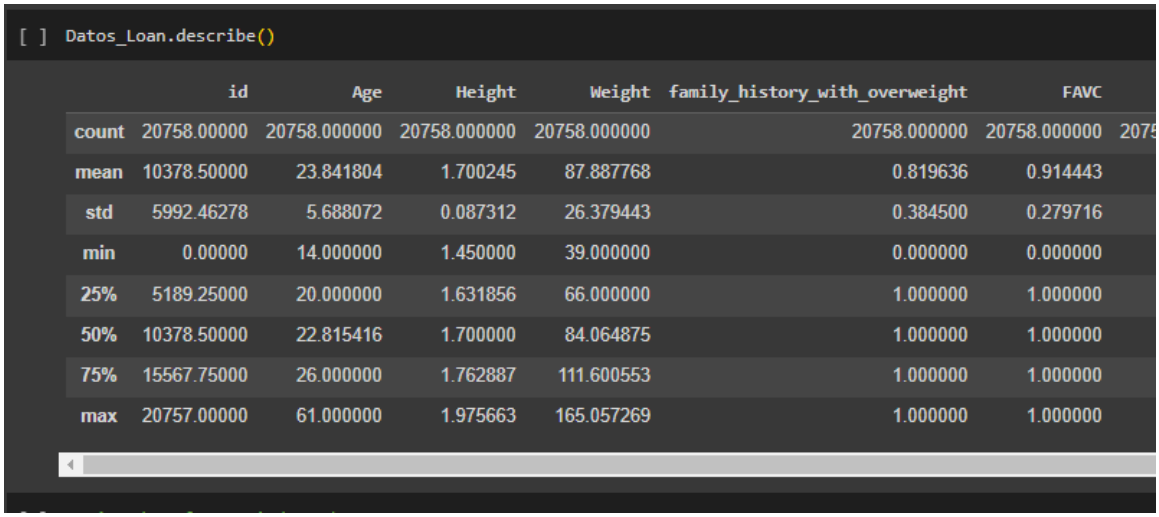
```
Datos_Loan.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20758 entries, 0 to 20757
Data columns (total 18 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              20758 non-null  int64
 1   Gender                          20758 non-null  object
 2   Age                             20758 non-null  float64
 3   Height                          20758 non-null  float64
 4   Weight                          20758 non-null  float64
 5   family_history_with_overweight  20758 non-null  int64
 6   FAVC                            20758 non-null  int64
 7   FCVC                            20758 non-null  float64
 8   NCP                             20758 non-null  float64
 9   CAEC                            20758 non-null  object
 10  SMOKE                           20758 non-null  int64
 11  CH2O                            20758 non-null  float64
 12  SCC                             20758 non-null  int64
 13  FAF                             20758 non-null  float64
 14  TUE                             20758 non-null  float64
 15  CALC                            20758 non-null  object
 16  MTRANS                          20758 non-null  object
 17  0be1dad                         20758 non-null  object
dtypes: float64(8), int64(5), object(5)
memory usage: 2.9+ MB

Conectado a
```

Figura 13. Información de los datos. Fuente: Elaboración propia.

En este caso (figura 13) se puede observar de forma muy resumida que contamos con, 17 etiquetas que cada una cuentan con 20758 datos, las cuales son 8 tipo float, 5 tipo int y 5 tipo object.

Descripción de los datos 3.1

Luego vamos a importar la función de Datos_Loan.describe() (figura 14) el cual consta de dar datos estadísticos. En ella se mostrarán los valores mínimos, valores máximos, la media, los porcentajes y la desviación estándar. Esta tabla nos describe de manera detallada si los datos están funcionando correctamente y no haya errores por arreglar. Además, desde esta tabla se hacen análisis en el caso de presentar un trabajo de lado más numérico o una visión estadística del algoritmo.



```
[ ] Datos_Loan.describe()
```

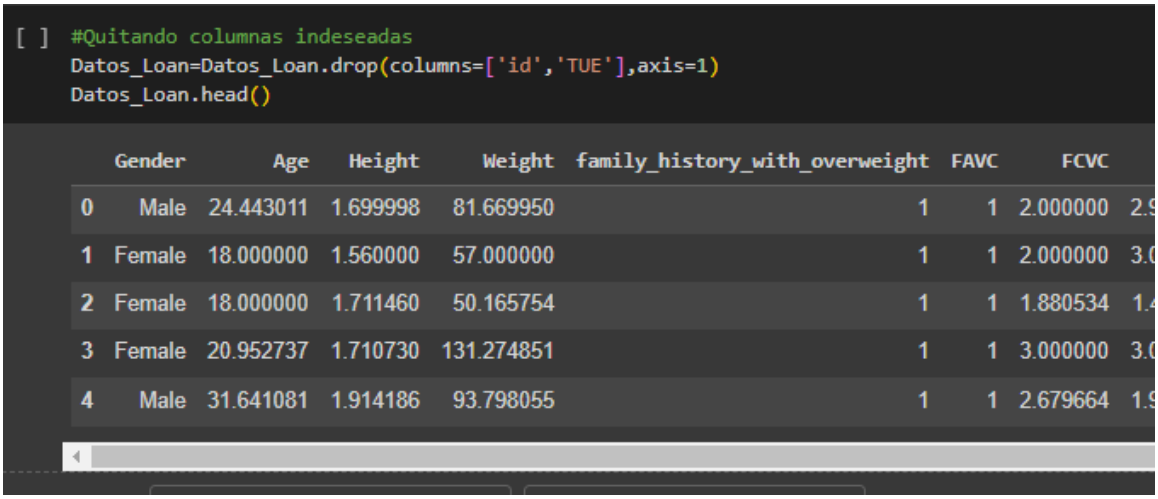
	id	Age	Height	Weight	family_history_with_overweight	FAVC	
count	20758.00000	20758.000000	20758.000000	20758.000000	20758.000000	20758.000000	2075
mean	10378.50000	23.841804	1.700245	87.887768	0.819636	0.914443	
std	5992.46278	5.688072	0.087312	26.379443	0.384500	0.279716	
min	0.00000	14.000000	1.450000	39.000000	0.000000	0.000000	
25%	5189.25000	20.000000	1.631856	66.000000	1.000000	1.000000	
50%	10378.50000	22.815416	1.700000	84.064875	1.000000	1.000000	
75%	15567.75000	26.000000	1.762887	111.600553	1.000000	1.000000	
max	20757.00000	61.000000	1.975663	165.057269	1.000000	1.000000	

Figura 14. Descripción estadística de los datos. Fuente: elaboración propia.

Eliminación de datos indeseadas 4.1

Como siguiente paso, quitaremos las columnas indeseadas. (figura 15) esto principalmente se hace cuando se considera que hay datos no relevantes o que no hacen un aporte a el enfoque al cual se llevar este análisis. En este caso hemos quitado dos

datos que no consideramos relevantes. Como lo son el id y TUE (Tiempo de uso de dispositivos tecnológicos). Ya que nuestro enfoque principal es determinar que niveles de obesidad puede tener una persona al ingresas sus datos.



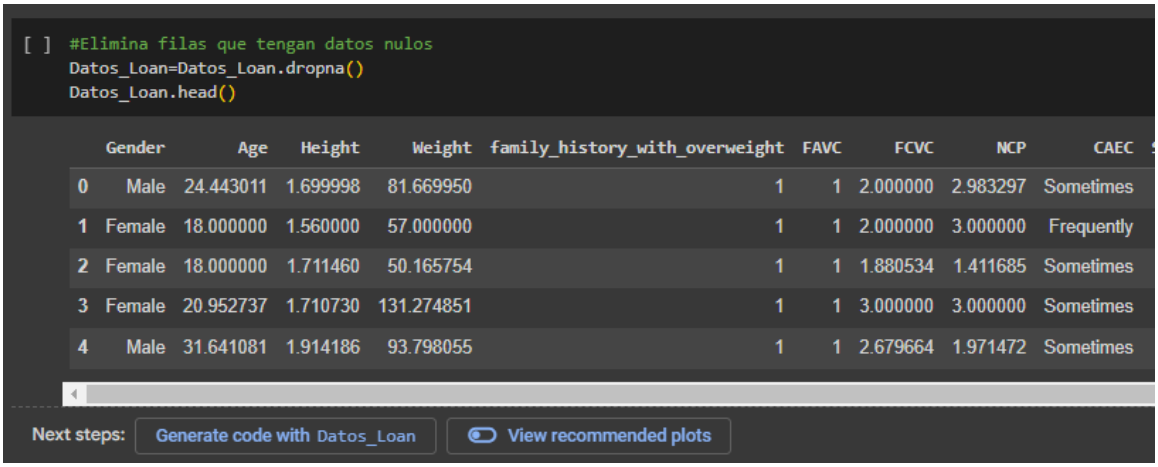
```
[ ]  #Quitando columnas indeseadas
     Datos_Loan=Datos_Loan.drop(columns=['id','TUE'],axis=1)
     Datos_Loan.head()
```

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	
0	Male	24.443011	1.699998	81.669950	1	1	2.000000	2.9
1	Female	18.000000	1.560000	57.000000	1	1	2.000000	3.0
2	Female	18.000000	1.711460	50.165754	1	1	1.880534	1.4
3	Female	20.952737	1.710730	131.274851	1	1	3.000000	3.0
4	Male	31.641081	1.914186	93.798055	1	1	2.679664	1.9

Figura 15. Eliminación de columnas indeseadas. Fuente: Elaboración propia.

Eliminación de datos nulos 5.1

Al momento de ser analizados los datos, es importante no tener en ellos valores nulos. Considerando que se hará un aprendizaje automatizado, estos modelos los cuales harán sus categorizaciones con respecto a la información que se le brindará, se deberá tener en cuenta la importancia de mantener valores correctos para que no afecte en sus tomas de decisiones y no entreguen predicciones incorrectas. (figura 16)



```
[ ]  #Elimina filas que tengan datos nulos
     Datos_Loan=Datos_Loan.dropna()
     Datos_Loan.head()
```

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC
0	Male	24.443011	1.699998	81.669950	1	1	2.000000	2.983297	Sometimes
1	Female	18.000000	1.560000	57.000000	1	1	2.000000	3.000000	Frequently
2	Female	18.000000	1.711460	50.165754	1	1	1.880534	1.411685	Sometimes
3	Female	20.952737	1.710730	131.274851	1	1	3.000000	3.000000	Sometimes
4	Male	31.641081	1.914186	93.798055	1	1	2.679664	1.971472	Sometimes

Next steps: Generate code with Datos_Loan View recommended plots

Figura 16. Eliminación de filas con valores nulos. Fuente: Elaboración Propia.

Análisis de niveles de estudio 6.1

Este paso es primordial para el desarrollo de este análisis. Como primer paso se hace la declaración a los datos que consideremos conocer. Es decir, cada columna consta de varios datos como se ha hablado en temas anteriores y que además estos datos han sido seleccionados por diferentes usuarios (figura 17,18,19,20). Uno ejemplo claro para contextualizar es en el caso de la columna genero (Gender)(figura 17). Esta columna se elige para ser analizada y por ende esta brindara la información que la contiene. En este caso solo contamos con dos valores los cuales son, el género femenino “Female” y el género masculino “Male”. De esta manera, así como en los análisis siguientes a este que se realizaron, arrojaron su información para conocerlos cada uno. A partir de esto, cuando son datos numéricos, se busca conocer el límite de estos datos debido a que los modelos aprenderán solo esos límites y al momento de conocer nuevos valores, Esto podría llevar a tener dificultades en cada predicción.

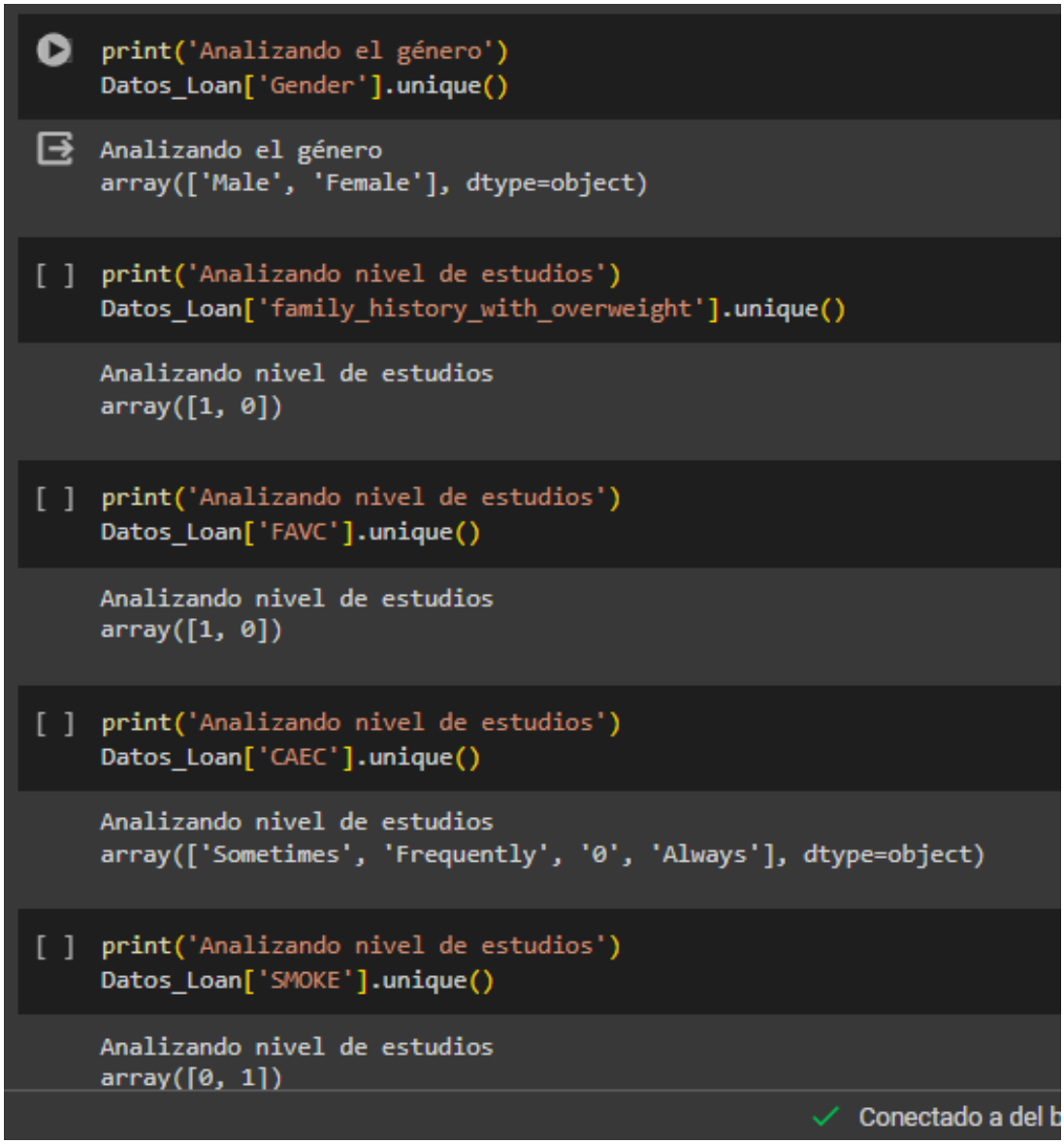


Figura 17. Análisis de niveles de estudio 1. Fuente: Elaboración propia.

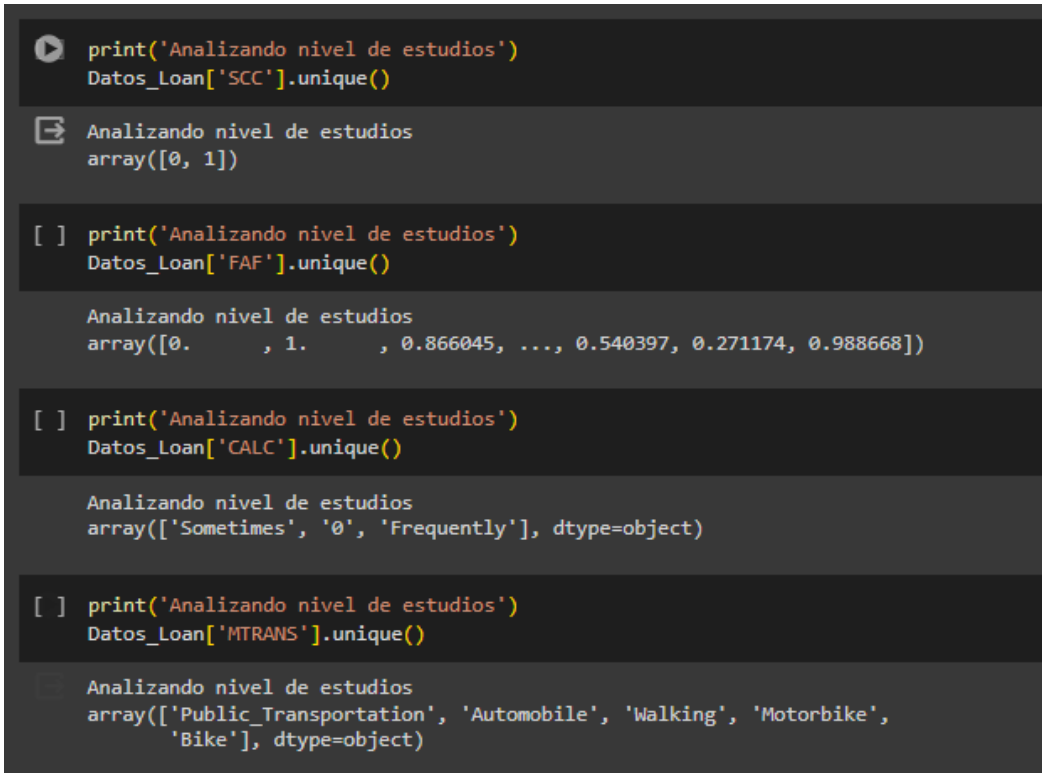


Figura 18. Análisis de niveles de estudio 2. Fuente: Elaboración propia.

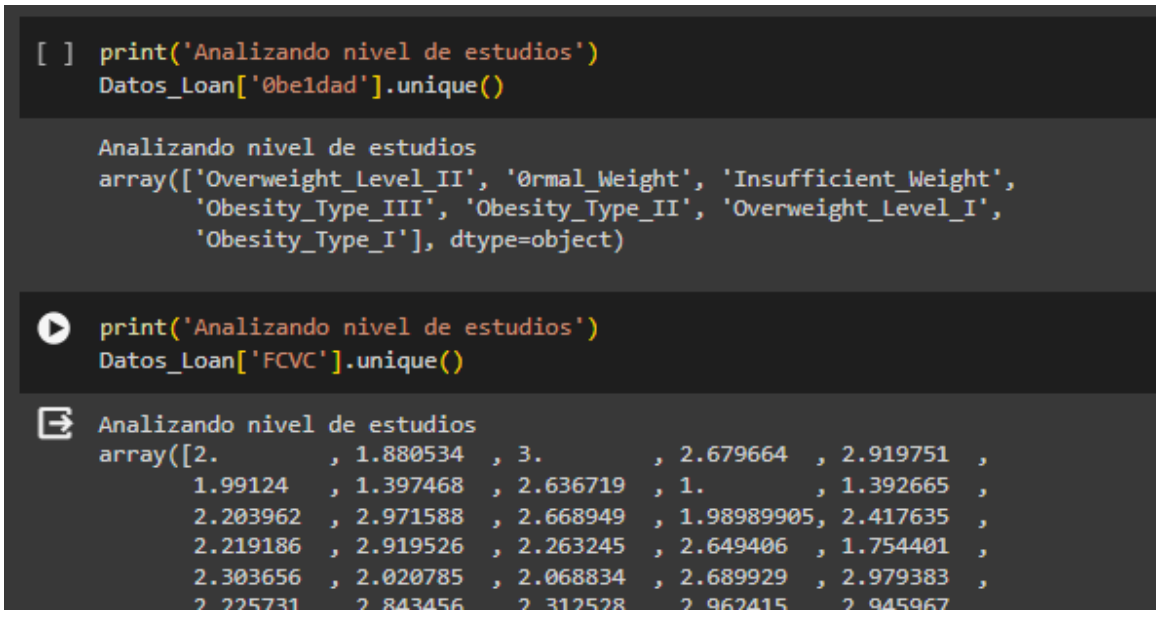


Figura 19. Análisis de niveles de estudio 3. Fuente: Elaboración propia.

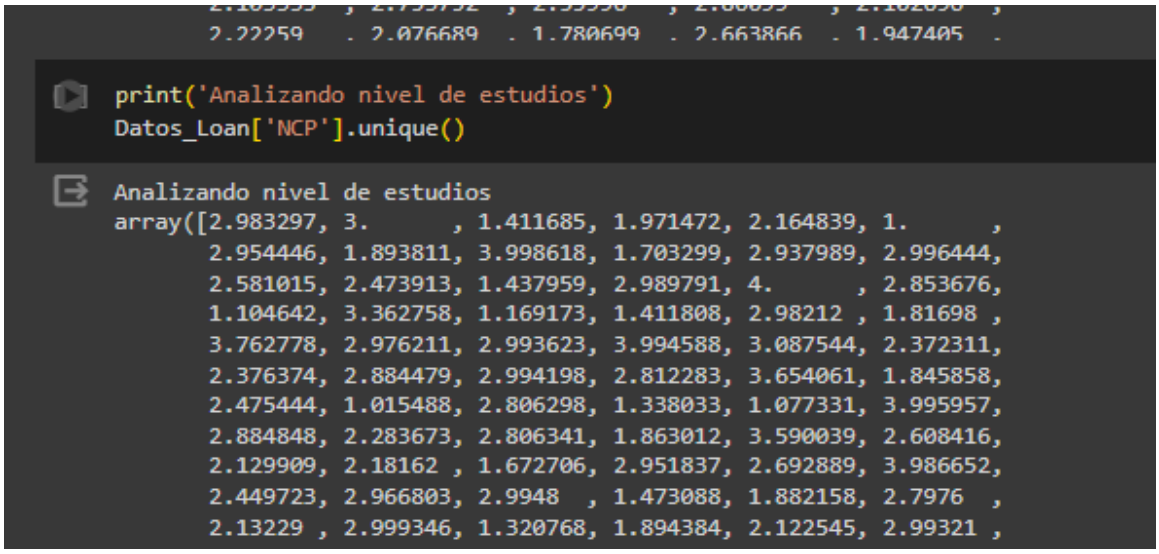


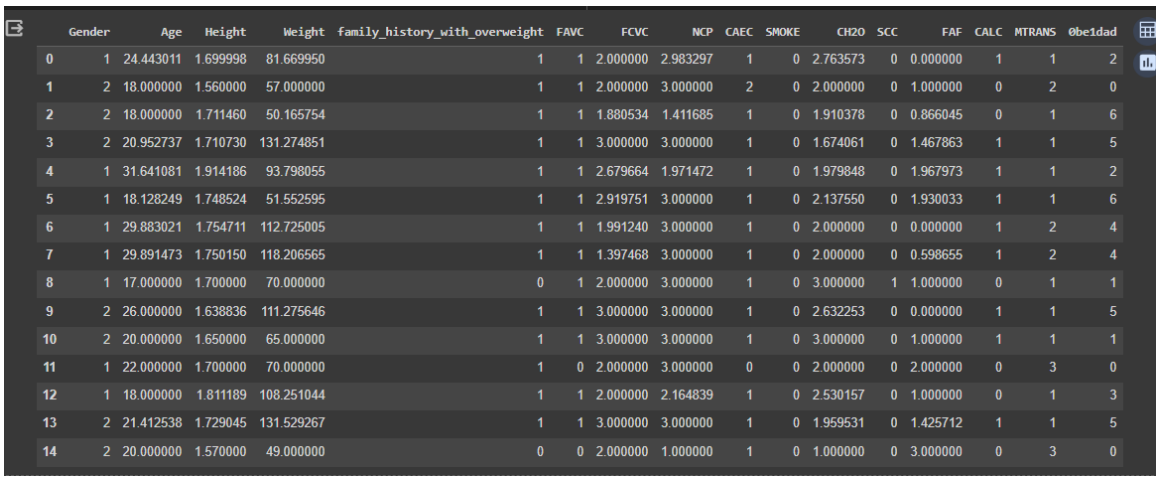
Figura 20. Análisis de niveles de estudio 4. Fuente: Elaboración propia.

Conversión de datos a números 7.1

cuando queremos manejar predicciones es fundamental los valores numéricos, por esta razón se han elegido ciertos datos los cuales originalmente son datos tipo objeto y que se convertirán tipo entero. De esta manera, se han elegido 5 remplazos, los cuales son, genero, obesidad, consumo de alcohol y medio de transporte (figura 21). Estos datos los hemos cambiado a números, los cuales se han decidido por elección propia y que agilizaran tanto las predicciones finales como la toma de decisiones para cada modelo.

Figura 21. conversión de datos a números. Fuente: Elaboración propia.

a comparación de la tabla que se presentó en un principio, se puede visualizar (figura 22) como se encuentra la tabla después de todos los cambios ya realizados anteriormente.



	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	CALC	MTRANS	0be1dad
0	1	24.443011	1.699998	81.669950	1	1	2.000000	2.983297	1	0	2.763573	0	0.000000	1	1	2
1	2	18.000000	1.560000	57.000000	1	1	2.000000	3.000000	2	0	2.000000	0	1.000000	0	2	0
2	2	18.000000	1.711460	50.165754	1	1	1.880534	1.411685	1	0	1.910378	0	0.866045	0	1	6
3	2	20.952737	1.710730	131.274851	1	1	3.000000	3.000000	1	0	1.674061	0	1.467863	1	1	5
4	1	31.641081	1.914186	93.798055	1	1	2.679664	1.971472	1	0	1.979848	0	1.967973	1	1	2
5	1	18.128249	1.748524	51.552595	1	1	2.919751	3.000000	1	0	2.137550	0	1.930033	1	1	6
6	1	29.883021	1.754711	112.725005	1	1	1.991240	3.000000	1	0	2.000000	0	0.000000	1	2	4
7	1	29.891473	1.750150	118.206565	1	1	1.397468	3.000000	1	0	2.000000	0	0.598655	1	2	4
8	1	17.000000	1.700000	70.000000	0	1	2.000000	3.000000	1	0	3.000000	1	1.000000	0	1	1
9	2	26.000000	1.638836	111.275646	1	1	3.000000	3.000000	1	0	2.632253	0	0.000000	1	1	5
10	2	20.000000	1.650000	65.000000	1	1	3.000000	3.000000	1	0	3.000000	0	1.000000	1	1	1
11	1	22.000000	1.700000	70.000000	1	0	2.000000	3.000000	0	0	2.000000	0	2.000000	0	3	0
12	1	18.000000	1.811189	108.251044	1	1	2.000000	2.164839	1	0	2.530157	0	1.000000	0	1	3
13	2	21.412538	1.729045	131.529267	1	1	3.000000	3.000000	1	0	1.959531	0	1.425712	1	1	5
14	2	20.000000	1.570000	49.000000	0	0	2.000000	1.000000	1	0	1.000000	0	3.000000	0	3	0

Figura 21. Tabla final. Fuente: Elaboración propia.

Modelo de toma de decisiones

En los siguientes pasos, se explicará todos los procesos por los cuales se deben tener en cuenta para el entrenamiento de los modelos con el fin de obtener toma de decisiones de forma correcta.

División de entradas y salidas 1.1.

En este paso es importante conocer los rangos o longitudes que estaremos manejando. por esto se ha decidido que, en valores de entrada contaremos con 15 datos, la cuales serán los datos que el usuario ingresara para obtener su diagnóstico. Por otro lado, tenemos los datos de salida, el cual se ha seleccionado solo 1 dato. claramente este será solo uno, debido a que el modelo solo dirá una respuesta según su predicción. A los datos adquiridos. (figura 23)

```
import numpy as np
Datos_matriz=np.array(Datos_Loan)

X = Datos_matriz[:,0:15] #datos de entrada (Todas las variables del cliente)
Y = Datos_matriz[:,-1] #Datos de salida (La decisiòn del nivel de obesidad)
```

Figura 23. División en entradas y salidas. Fuente: Elaboración propia.

División de datos de entrenamiento y validación 2.1

la división consta de partir estos datos de la siguiente forma (figura 24) con el fin de evitar cualquier problema que se llegue a presentar ante el momento de realizar los entrenamientos al algoritmo y ante la toma de decisiones de este. su función claramente es darle a conocer a el algoritmo sus datos correspondientes.

```
import sklearn
from sklearn.model_selection import train_test_split
X_train, X_test,Y_train, Y_test= train_test_split(X,Y,test_size=0.1,random_state=751)
```

Figura 24. División de datos de entrenamiento y validación. Fuente: Elaboración propia.

```
#Para mejorar la escala de los datos se hace normalization (Ignorar)
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Figura 25. Normalización de datos. Fuente: Elaboración propia

Además, contamos con una normalización de los datos, la cual está hecha con la finalidad de ajustar valores que se encuentren nulos o duplicados (figura 25). Para así, brindarles a los modelos de predicción características claras y puedan desarrollar mejor sus predicciones.

Evaluación de casos mediante todos los modelos de predicciones 3.1

De esta manera es como se entrenan todos los modelos que hemos seleccionado (figura 26) y que de los cuales obtendremos una toma de decisión según su forma de clasificar los datos. Para comenzar en la parte superior se hace el llamado de los modelos que utilizaremos. cómo se presentó anteriormente, contamos con el modelo de KNN, Bayes, LDA y SVM. Estos cuatro modelos que serán los encargados de direccionar a la siguiente persona que ingrese sus datos en el algoritmo.

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score,precision_score

Modelo_0 = KNeighborsClassifier(5)
Modelo_0.fit(X_train, Y_train)
Y_pred_0 =Modelo_0.predict (X_test)
print("Accuracy KNN",accuracy_score(Y_test, Y_pred_0))

Modelo_1 = GaussianNB()
Modelo_1.fit(X_train, Y_train)
Y_pred =Modelo_1.predict (X_test)
print("Accuracy Bayes",accuracy_score(Y_test, Y_pred))

Modelo_2 = LinearDiscriminantAnalysis()
Modelo_2.fit(X_train, Y_train)
Y_pred_2 =Modelo_2.predict (X_test)
print("Accuracy LDA",accuracy_score(Y_test, Y_pred_2))

Modelo_3 = SVC()
Modelo_3.fit(X_train, Y_train)
Y_pred_3 =Modelo_3.predict (X_test)
print("Accuracy SVM",accuracy_score(Y_test, Y_pred_3))
```

Figura 26. Evaluando casos mediante todos los clasificadores. Fuente: Elaboración propia.

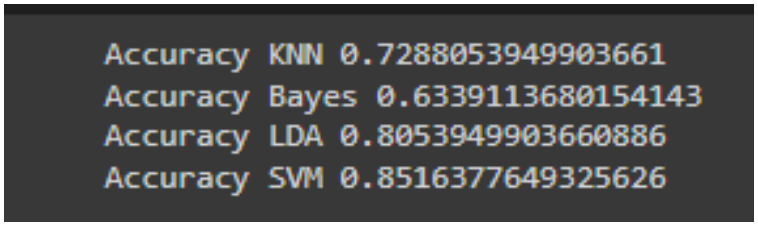


Figura 27. Resultados de los clasificadores. Fuente: Elaboración propia.

Se puede observar (figura 27), como los clasificadores muestran sus resultados del rango de valores de probabilidad. Estos números muestran la confianza que pueden tener los usuarios a este modelo, en este caso el rango esta entre 0 y 1.

Haciendo un análisis breve, los modelos cuentan con muy buen limite de confianza, ya que se acercan mucho al número 1.

Probando los modelos entrenados 4.1

Para comenzar a probar los modelos entrenados, es fundamental implementar todas las preguntas que se consideran pertinentes para el desarrollo del análisis y su debida predicción (figura 28). En este caso se contaron con 15 preguntas, las cuales su enfoque principal es identificar si el usuario que esta ingresando los datos, cuenta con los mismos hábitos o se hacen semejante a los que cuenta el dataframe para obtener su predicción e indicarle en qué nivel de obesidad se encuentre. (figura 28) los datos ingresados se normalizan con el fin interpretarlos de una manera mas adecuada, agilizando los procesamientos y buscando sus mejores respuestas. (figura 29)

```
#Probando el modelo entrenado sobre un nuevo sujeto
Target=np.zeros((1,15))
Target[0,0]=float(input('Ingrese género, 1 para Masculino y 2 para Femenino: '))
Target[0,1]=float(input('Ingrese su edad: '))
Target[0,2]=float(input('Ingrese su atura: '))
Target[0,3]=float(input('Ingrese su peso: '))
Target[0,4]=float(input('¿tiene familiares con obesidad?, 1 para si y 0 para no: '))
Target[0,5]=float(input('¿Consumo frecuente de alimentos ricos en calorías?, 1 para si y 0 para no: '))
Target[0,6]=float(input('Consumos frecuentes de vegetales, entre 1 y 3: '))
Target[0,7]=float(input('Cuantos alimentos principales consume, entre 1 y 3: '))
Target[0,8]=float(input('¿consume alimentos entre comidad?, 0 para no, 1 para aveces, 2 frecuentemente y 3 siempre: '))
Target[0,9]=float(input('¿Fuma?, 0 para no, 1 para si: '))
Target[0,10]=float(input('¿cuantos litros toma al dia?, entre 1 y 3:  '))
Target[0,11]=float(input('¿Usted toma bebidas caloricas?, 0 para no, 1 si: '))
Target[0,12]=float(input('¿hace actividad fisica con frecuencia?, 0 para no, 1 para si: '))
Target[0,13]=float(input('¿consume bebidad alcoholicas?, 0 para no, 1 para aveces, 2 frecuentemente: '))
Target[0,14]=float(input('¿cual es su medio de traspoerte?, 1 para trasporte publico, 2 auto, 3 camiando, 4 moto, 5 cicla: '))
```

Figura 28. Probando los modelos con nuevos datos. Fuente: elaboración propia.

```
Target = scaler.transform(Target) #Normalizar los datos

Prediction_0 =Modelo_0.predict (Target)
Prediction_1 =Modelo_1.predict (Target)
Prediction_2 =Modelo_2.predict (Target)
Prediction_3 =Modelo_3.predict (Target)
```

Figura 29. Normalización de los datos ingresados. Fuente: elaboración propia.

imprimir las predicciones 5.1

como último paso, se imprimen las predicciones, para que cada modelo pueda identificar las semejanzas y pueda dar sus predicciones. Este paso se hace con cada modelo, indicando los niveles que hay de obesidad para su toma de decisión. (figura 30, 31, 32, 33)

```
print(" ")

if Prediction_0==0:
  print("Según KNN, se encuentra en peso normal")
elif Prediction_0==1:
  print("Según KNN, se encuentra en Sobrepeso nivel I")
elif Prediction_0==2:
  print("Según KNN, se encuentra en Sobrepeso nivel II")
elif Prediction_0==3:
  print("Según KNN, se encuentra en Obesidad tipo I")
elif Prediction_0==4:
  print("Según KNN, se encuentra en Obesidad tipo II")
elif Prediction_0==5:
 print("Según KNN, se encuentra en Obesidad tipo III")
else:
  print("Según KNN, es un Peso insuficiente")

print(" ")
```

Figura 30. Predicción, modelo KNN. Fuente: Elaboración propia

```
if Prediction_1==0:
  print("Según bayes, se encuentra en peso normal")
elif Prediction_1==1:
  print("Según bayes, se encuentra en Sobrepeso nivel I")
elif Prediction_1==2:
  print("Según bayes, se encuentra en Sobrepeso nivel II")
elif Prediction_1==3:
  print("Según bayes, se encuentra en Obesidad tipo I")
elif Prediction_1==4:
  print("Según bayes, se encuentra en Obesidad tipo II")
elif Prediction_1==5:
  print("Según bayes, se encuentra en Obesidad tipo III")
else:
  print("Según bayes, es un Peso insuficiente")

print(" ")
```

Figura 31. Predicción, modelo bayes. Fuente: Elaboración propia

```
if Prediction_2==0:
  print("Según LDA, se encuentra en peso normal")
elif Prediction_2==1:
  print("Según LDA, se encuentra en Sobrepeso nivel I")
elif Prediction_2==2:
  print("Según LDA, se encuentra en Sobrepeso nivel II")
elif Prediction_2==3:
  print("Según LDA, se encuentra en Obesidad tipo I")
elif Prediction_2==4:
  print("Según LDA, se encuentra en Obesidad tipo II")
elif Prediction_2==5:
  print("Según LDA, se encuentra en Obesidad tipo III")
else:
  print("Según LDA, es un Peso insuficiente")

print(" ")
```

Figura 32. Predicción, modelo LDA. Fuente: Elaboración propia

```
if Prediction_5==0:
  print("Según SVM, se encuentra en peso normal")
elif Prediction_5==1:
  print("Según SVM, se encuentra en Sobrepeso nivel I")
elif Prediction_5==2:
  print("Según SVM, se encuentra en Sobrepeso nivel II")
elif Prediction_5==3:
  print("Según SVM, se encuentra en Obesidad tipo I")
elif Prediction_5==4:
  print("Según SVM, se encuentra en Obesidad tipo II")
elif Prediction_5==5:
  print("Según SVM, se encuentra en Obesidad tipo III")
else:
  print("Según SVM, es un Peso insuficiente")

print(" ")
```

Figura 33. Predicción, modelo SVM. Fuente: Elaboración propia

Resultado de las predicciones 6.1

Finalmente podemos observar los datos que ingresaron dos usuarios en cada pregunta, y sus predicciones. El cual cuenta un usuario con sobre peso y la otra se encuentra en peso normal (figura 34, 35).

```
Ingrese género, 1 para Masculino y 2 para Femenino: 2
Ingrese su edad: 50
Ingrese su atura: 1.78
Ingrese su peso: 89
¿tiene familiares con obesidad?, 1 para si y 0 para no: 0
¿Consumo frecuente de alimentos ricos en calorías?, 1 para si y 0 para no: 1
Consumos frecuentes de vegetales, entre 1 y 3: 3
Cuantos alimentos principales consume, entre 1 y 3: 3
¿consume alimentos entre comidad?, 0 para no, 1 para aveces, 2 frecuentemente y 3 siempre: 1
¿Fuma?, 0 para no, 1 para si: 0
¿cuantos litros toma al dia?, entre 1 y 3:  3
¿Usted toma bebidas caloricas?, 0 para no, 1 si: 0
¿hace actividad fisica con frecuencia?, 0 para no, 1 para si: 0
¿consume bebidad alcoholicas?, 0 para no, 1 para aveces, 2 frecuentemente: 1
¿cual es su medio de traspoerte?, 1 para trasporte publico, 2 auto, 3 camiando, 4 moto, 5 cicla: 3

Según KNN, se encuentra en peso normal

Según bayes, se encuentra en Sobrepeso nivel II

Según LDA, se encuentra en Sobrepeso nivel II

Según SVM, se encuentra en Sobrepeso nivel I
```

Figura 34. Resultados de las predicciones 1. Fuente: elaboración propia.

```
Ingrese género, 1 para Masculino y 2 para Femenino: 2
Ingrese su edad: 21
Ingrese su atura: 1.55
Ingrese su peso: 45
¿tiene familiares con obesidad?, 1 para si y 0 para no: 0
¿Consumo frecuente de alimentos ricos en calorías?, 1 para si y 0 para no: 1
Consumos frecuentes de vegetales, entre 1 y 3: 3
Cuantos alimentos principales consume, entre 1 y 3: 3
¿consume alimentos entre comidad?, 0 para no, 1 para aveces, 2 frecuentemente y 3 siempre: 1
¿Fuma?, 0 para no, 1 para si: 0
¿cuantos litros toma al dia?, entre 1 y 3:  1.2
¿Usted toma bebidas caloricas?, 0 para no, 1 si: 0
¿hace actividad fisica con frecuencia?, 0 para no, 1 para si: 0
¿consume bebidad alcoholicas?, 0 para no, 1 para aveces, 2 frecuentemente: 1
¿cual es su medio de traspoerte?, 1 para trasporte publico, 2 auto, 3 camiando, 4 moto, 5 cicla: 4

Según KNN, se encuentra en peso normal

Según bayes, se encuentra en peso normal

Según LDA, es un Peso insuficiente

Según SVM, se encuentra en peso normal
```

Figura 35. Resultados de las predicciones 2. Fuente: elaboración propia.

Implementación en contextos reales

En medios tecnológicos podemos observar como también hay algoritmos similares como el que hemos realizado. En primer lugar, tenemos una investigación la cual consta de un algoritmo de clasificación llamado predicción de casos de obesidad infantil [12] donde se habla de las predicciones tempranas de obesidad en niños de 3 años.

También contamos con otra detección de obesidad en este caso en adolescentes llamado Algoritmos de clasificación para la detección de obesidad en adolescentes: Un estudio comparativo entre KNN y árboles de decisión [13] donde se identifican la obesidad temprana en adolescentes entre 15 y 19 años.

Resultados adicionales

Dentro de las aplicaciones que se presentan como predicciones de diagnósticos, se puede desplegar un sistema de alertas en pacientes que quieran conocer su estado físico sin la necesidad ir a un centro médico. Primeramente, para poder realizar alertas se requiere la obtención de la información de personas que participen para la obtención de datos reales sobre su estado físico para ayudar a usuarios futuros cuando deben preocuparse por su estado de salud.

Conclusiones

Se llego a la conclusión que los modelos de predicción utilizados en el diseño e implementación del algoritmo de aprendizaje supervisado que hemos realizado, los cuales son KNN, Bayes, LDA y SVM lo cual nos proporcionó beneficios ya que han obtenido resultados óptimos con respecto a los datos ingresados sobre riegos de obesidad, y se comprueba su precisión ante las pruebas de una manera clara. También se reconoce que los modelos de predicción se deben entrenar de una manera adecuada y normalizada. El procesamiento de los datos es fundamental debido a que, si se encuentran valores nulos o incorrectos, afectaría al generar las predicciones. El análisis de los resultados se revelaron varios niveles de obesidad según su masa corporal (IMC) según estos datos se utilizaron de manera correcta y especifica para las predicciones futuras junto con los valores determinados por cada usuario que se encontró en la base de datos.
Básicamente este análisis nos lleva a determinar que la inteligencia artificial puede establecerse en cualquier campo, teniendo en cuenta este caso, sería crucial la implementación de este algoritmo para ayudar a profesionales de la salud con la finalidad de agilizar los resultados sobre riesgos de obesidad de las personas que están empezando o como a las personas que ya se encuentran en un estado de sobrepeso alto.

Referencias

Referencias

[1] Martinez, J. A., Moreno-Aliaga, M. J., Marques-Lopes, I., & Marti, A. (2002). Causas de obesidad.

[2] Basulto, J., Manera, M., Baladia, E., Miserachs, M., Pérez, R., Ferrando, C., ... & Revenga, J. (2013). Definición y características de una alimentación saludable. Monografia a Internet].

[3] Bascon, M. A. P. (1994). Actividad física y salud. Obtenido de https://archivos. csif. es/archivos/andalucia/ensenanza/revistas/csicsif/revista/pdf/Numero_42/MIGUEL_ANGEL_PRIETO_BASCON_01. pdf.

[4] Google Colab. (s. f.). Recuperado 29 de marzo de 2024, de https://research.google.com/colaboratory/intl/es/faq.html#whats-colaboratory.

[5] Aransay, J., Casado-García, Á., Domínguez, C., García-Domínguez, M., Heras, J., Inés, A., ... & Pérez, B. (2022). GitHub y Google Colaboratory para el desarrollo, comunicación y gestión de prácticas en los laboratorios de informática.

[6] García, A., Martínez, G., Nuñez, E., & Guzmán, A. (1998). Clasificación supervisada, inducción de arboles de decisión, algoritmo kd. Proc. Simp. Int. de Comp. CIC, 98, 602-614.

[7] Obesity Risk Dataset. (2024, 11 marzo). Kaggle. https://www.kaggle.com/datasets/jpkochar/obesity-risk-dataset

[8] Madariaga Fernández, C. J., Lao León, Y. O., Curra Sosa, D. A., & Lorenzo Martín, R. (2022). Empleo de algoritmos KNN en metodología multicriterio para la clasificación de clientes, como sustento de la planeación agregada. Retos de la Dirección, 16(1), 178-198.

[9] Fuster Coma, N. (2023). Métodos de Clasificación en Python: Aplicaciones a la Empresa (Doctoral dissertation, Universitat Politècnica de València).

[10] Balakrishnama, S. y Ganapathiraju, A. (1998). Análisis discriminante lineal: un breve tutorial. Instituto de Procesamiento de Señales e Información , 18 (1998), 1-8.

[11] Qisthiano, MR, Ruswita, I. y Prayesy, PA (2023). Método de implementación de SVM en el análisis de sentimientos que se utilizan con el uso de Python 3. Tecnología: Jurnal Ilmiah Sistem Informasi , 13 (1), 1-7.

[12] Suca, C., Córdova, A., Condori, A., Cayra, J., & Sulla, J. (2016). Comparación de algoritmos de clasificación para la predicción de casos de obesidad infantil. Perú: Universidad Nacional de San Agustín.

[13] Díaz, S. E. L., Sibaja, J. A. P., Martínez, A. F., & Vázquez, S. J. (2023). Algoritmos de clasificación para la detección de obesidad en adolescentes: Un estudio comparativo entre KNN y árboles de decisión. Revista de Investigación en Tecnologías de la Información, 11(23), 70-81.