Matemáticas y ciencia de datos para la toma de decisiones

Evidencia 2: Proyecto de Ciencia de Datos

Realizado por:

Alejandro Paredes Balgañón - A01351746

Tecnológico de Monterrey campus Irapuato

30 de mayo del 2021

Introducción

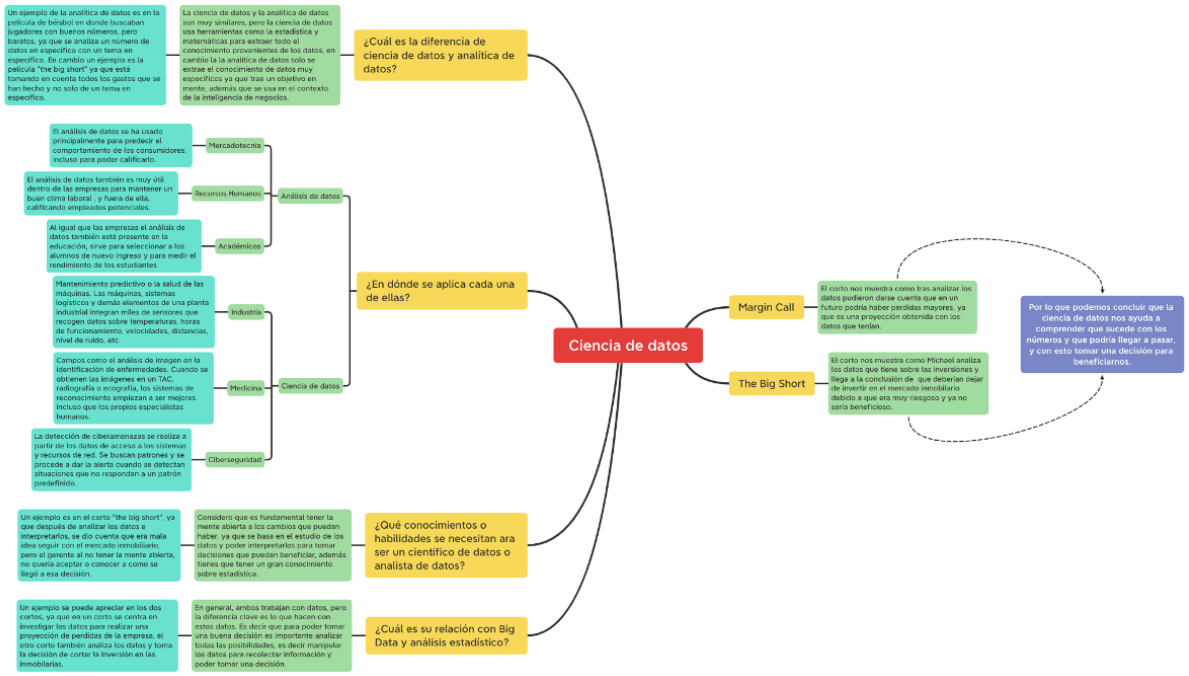
La Ciencia de Datos se encarga de analizar grandes cantidades de información con la ayuda de la inteligencia artificial para mejorar el manejo de la información.
Se trata de combinar las técnicas de ciencia de la computación y la estadística, para tener un reconocimiento de patrones, generar modelos de probabilidad y la visualización, para analizar e interpretar datos. Con esto, la ciencia de datos puede descubrir tendencias, predecir el futuro y anticiparlo por medio de las probabilidades.

La ciencia de datos puede ayudar a una persona en diferentes cosas, estas cosas pueden ser metas que uno se proponga, como lo puede ser el ahorrar dinero. Esta meta se puede lograr al crear una base de datos con lo relacionado al dinero, es decir tener de variables el dinero que ganas, el dinero que gastas, y por medio de esto podrás sacar el dinero que te queda, además que puedes agregar más variables para conocer el comportamiento del dinero que tienes. Y este es un ejemplo de muchos en donde la ciencia de datos puede ayudar a las personas a tomar decisiones más acertadas para tener un mayor beneficio.

En el caso de este proyecto, su intención es generar una base de datos con lo que consumimos, para que a partir de los datos recolectados se puedan analizar y tomar una decisión dependiendo el resultado obtenido, es decir, que gracias a este proyecto nos podremos dar una idea de si estamos bien alimentados, o tenemos menos calorías de las necesarias, o viceversa, es decir, tenemos más calorías de las necesarias. En nuestro caso se analizaron 5 variables, las calorías como variable dependiente, y como variable independientes los nutrientes, carbohidratos, lípidos, proteínas y sodio de cada comida que consumimos durante este semestre.

Fase 1: Entendimiento del negocio

Mapa mental



Las etapas de esta fase son:

• Identificación de los objetivos del negocio.

Esta fase consiste en identificar una meta u objetivo que se tiene planeado lograr, en nuestro proyecto de poner las calorías de los alimentos que consumimos, nos serviría para plantear un objetivo como comer más saludable o disminuir un tipo de comida, por lo que es fundamental conocer que se necesita para lograrlo.

• Evaluación de la situación.

Para evaluar la situación es importante ya tener bien identificado el objetivo que queremos lograr, ya que debemos tomar en cuenta como empezamos, es decir tenemos que crear varias hipótesis sobre la problemática que estamos tratando, como por ejemplo si mi objetivo es tener una buena salud, al evaluar la situación debo preguntarme que me falta para tener la buena salud, por lo que nuestras hipótesis podrían ser tener una buena alimentación o la de practicar algún deporte.

• Definición de los objetivos para la analítica

minería de datos.

En este punto debemos saber que datos queremos recolectar, ya que estos nos ayudarán a analizar y ver que comportamiento o patrones existen, como por ejemplo el consumo de calorías, y nutrientes de los alimentos que consumo, esta definición

exacta de que datos quiero recolectar, me ayudará más tarde a armar un plan de trabajo y a tomar una decisión dependiendo que resultado tenga.

• Desarrollo de un plan de trabajo

Una vez conociendo el objetivo y tener muy en claro que datos quiero recolectar es tiempo de generar un plan de trabajo que permita recolectar esta información, como lo fue en nuestro caso de crear un excel en donde ingresamos los datos de los alimentos que consumimos diariamente con sus respectivas calorías y nutrientes.

1. ¿Quién es el cliente?

El cliente en este proyecto soy yo, ya que la base de datos de los alimentos que he consumido me servirá para tomar alguna decisión sobre si debo mejorar mi alimentación.

2. ¿Qué problemas estás tratando de resolver?

El problema que está tratando de resolver es uno que en México es muy común, el cual es la obesidad, ya que esta base de datos nos estará dando información de cuantas calorías y nutrientes consumimos.

3. ¿Qué solución o soluciones la Ciencia de Datos tratará de proveer?

La ciencia de datos nos ayudará para entender los datos que recolectamos, como lo puede ser para generar un modelo matemático para conocer como nos afecta o conocer cuantas calorías consumimos y si nos faltan o debemos hacer algo para disminuirlas.

4. ¿Qué necesitas aprender para poder desarrollar la solución o soluciones?

Para poder generar una solución o tomar una decisión, debería conocer más a como manipular los dato y como interpretarlos, es decir aumentar mis conocimientos sobre la ciencia de datos, ya que engloba la estadística, matemáticas, entre otras, una vez conociendo esto y sabiendo si debo mejorar mi alimentación, debo asistir con un nutriologo para que pueda asesorarme de los alimentos que debo consumir.

5. ¿Qué deberás hacer para desarrollar tu solución?

Para desarrollar mi solución debo seguir aprendiendo sobre la ciencia de datos, por lo que seguiré haciendo las actividades e investigando sobre el tema para ampliar mis conocimientos, además de que seguiré plasmando los alimentos que consumo diario sin modificarlos para generar la solución de la realidad y no solo generar un trabajo para pasar la materia.

En resumen este trabajo es para que pueda saber como funciona la ciencia de datos y como puede beneficiarme en mi día a día, si bien este trabajo se centra en la alimentación, no implica que no pueda usarlo en otras cosas, al contrario podría usarlo para ahorrar, es decir en vez de tomar calorías y los nutrientes, puedo tomar en cuenta las variables de dinero ahorrado y dinero gastado para conocer cuando dinero tengo, es decir que la ciencia de datos cada vez se está haciendo más importante porque tanto nosotros, como las empresas, buscan conocer el comportamiento de diferentes cosas por medio de los datos para tomar la mejor decisión.

Fase 2. Entendimiento de los datos

Parte 1: Cargando mis datos en Python

```
import pandas as pd # importa la librería pandas y la asigna a la variable pd
datos_consumo= pd.read_excel('A01351746excel.xlsx') # indicamos el nombre de nuestro archivo a ser leído
datos_consumo.head() #comprobar que los datos se cargaron correctemente en el dataframe viendo los primeros 5 registros
```

	Fecha (dd/mm/aa)	Momento	Nombre alimento	Calorías (kcal)	Carbohidratos (g)	Lípidos/grasas (g)	Proteína (g)	Sodio (mg)	Fuente	Unnamed: 9	Unnamed: 10
0	2021-02-14	desayuno	Consome de barbacoa	238	1.49	14.18	25.34	834	fatsecret	NaN	Matrícula:
1	2021-02-14	comida	hamburguesa	552	35.46	29.90	33.17	994	fatsecret	NaN	Nombre:
2	2021-02-14	cena	hotdog	284	22.85	16.59	10.24	919	fatsecret	NaN	Fecha:
3	2021-02-15	desayuno	3 salchichas de pavo	465	1.41	24.24	56.37	1779	fatsecret	NaN	Momento de consumo:
4	2021-02-15	comida	3 tostadas de pollo	426	29.52	21.15	30.33	606	fatsecret	NaN	NaN

```
[5] datos_consumo.shape #conocer la forma, total de filas y columnas de nuestros datos

(203, 12)
```

```
[6] datos_consumo.columns #podemos ver los nombres de todas las columnas o atributos de nuestros datos

Index(['Fecha (dd/mm/aa)', 'Momento', 'Nombre alimento', 'Calorías (kcal)',
       'Carbohidratos (g)', 'Lípidos/grasas (g)', 'Proteína (g)', 'Sodio (mg)',
       'Fuente', 'Unnamed: 9', 'Unnamed: 10', 'Unnamed: 11'],
      dtype='object')
```

```
datos_consumo.dtypes #podemos conocer los tipos de datos de nuestro dataframe

Fecha (dd/mm/aa)      datetime64[ns]
Momento                       object
Nombre alimento               object
Calorías (kcal)                int64
Carbohidratos (g)            float64
Lípidos/grasas (g)           float64
Proteína (g)                 float64
Sodio (mg)                     int64
Fuente                        object
Unnamed: 9                   float64
Unnamed: 10                   object
Unnamed: 11                   object
dtype: object
```

```
datos_consumo.info() #nos regresa la información completa de los datos del dataframe

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 203 entries, 0 to 202
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Fecha (dd/mm/aa)    203 non-null    datetime64[ns]
 1   Momento             203 non-null    object
 2   Nombre alimento     203 non-null    object
 3   Calorías (kcal)     203 non-null    int64
 4   Carbohidratos (g)   203 non-null    float64
 5   Lípidos/grasas (g)  203 non-null    float64
 6   Proteína (g)        203 non-null    float64
 7   Sodio (mg)          203 non-null    int64
 8   Fuente              203 non-null    object
 9   Unnamed: 9          0 non-null      float64
 10  Unnamed: 10         8 non-null      object
 11  Unnamed: 11         14 non-null     object
dtypes: datetime64[ns](1), float64(4), int64(2), object(5)
memory usage: 19.2+ KB
```

```
[9] datos_consumo.describe()#para obtener la estadística descriptiva y completar la exploración de nuestros datos
```

	Calorías (kcal)	Carbohidratos (g)	Lípidos/grasas (g)	Proteína (g)	Sodio (mg)	Unnamed: 9
count	203.000000	203.000000	203.000000	203.000000	203.000000	0.0
mean	328.389163	28.664877	16.030542	17.638719	611.477833	NaN
std	212.200590	22.296163	13.077159	17.926805	636.125636	NaN
min	12.000000	0.000000	0.000000	0.000000	0.000000	NaN
25%	162.000000	12.600000	6.700000	2.300000	115.000000	NaN
50%	284.000000	24.000000	15.600000	10.880000	350.000000	NaN
75%	465.000000	43.155000	24.240000	29.520000	975.000000	NaN
max	1192.000000	92.000000	70.360000	63.480000	2757.000000	NaN

Parte 2: Describiendo mis datos

La fase dos se basa en la recolección de datos, es decir en los datos adquiridos, que tienen la posibilidad de poder complementar los datos con información externa con el objetivo de poder enriquecer el análisis. Los datos existentes que incluye una amplia variedad de datos, como datos transaccionales, datos de encuestas, registros web, y por último los datos adicionales que busca complementar la información ya adquirida si fuera necesario.

La creación de datos visuales ayudan mucho a la hora de explicar para que más gente pueda comprenderlo, la presentación efectiva de los resultados cuantitativos es una técnica que se ha utilizado durante mucho tiempo, este es lo que se conoce como visualización de datos. La visualización de datos debe cubrir el esquema del proceso de investigación, el resumen y las implicaciones de los resultados, así como la recomendación para la acción, ya que entre los analistas más efectivos se encuentran aquellos que pueden contar una historia con datos, pero las buenas historias presentan los hallazgos en términos que el público pueda entender.

En resumen la presentación de los datos debe presentar todo el procedimiento que se llevó a cabo para llegar a la solución, además de tener apoyo visual para mejorar la calidad de la presentación.

¿Cuáles son tus datos existentes (registrados), datos adquiridos (datos externos) y datos adicionales (datos generados)?
Los existentes son los datos que están en la base de datos, los datos adquiridos fueron las calorías y nutrientes sacados de una calculadora de nutrientes, y los datos adicionales son los generados a partir de estos, como gráficas, las medidas de dispersión o de tendencia central, etc.
¿Qué tipos de datos se analizarán?
Los datos a analizar serán los nutrientes y calorías que tienen las comidas que se consume al día, es decir que mi base de datos será de las comidas que yo consuma, por lo que será distinto a la de mis compañeros de materia.
¿Qué atributos (columnas) de la base de datos parecen más prometedores?
Las calorías, ya que esta me dice de manera general que tan bien me alimento.
¿Qué atributos parecen irrelevantes y pueden ser excluidos?
En lo personal siento que todo es importante, ya que la fecha te ayuda a conocer si hay un progreso o lo contrario, dividir los alimentos en comida, snack, y cena también te ayuda a generar un análisis más profundo en uno de estos tres, pero si deseas hacerlo general podría ser el que más se podría omitir, y por otra parte los nutrientes y calorías es lo que estamos analizando.
¿Hay datos suficientes (filas) para sacar conclusiones generalizables o hacer predicciones precisas?

Si, considero que ya hay una cantidad de datos para ir conociendo su tipo de alimentación, por lo que se podría tomar decisiones a partir de los datos ya recabados.

¿Hay demasiados atributos para realizar un modelo que sea fácil de interpretar?

No, considero que no hay demasiados atributos, ya que los seleccionados aportan información valiosa y que nos podría ayudar a tomar una mejor decisión.

¿De dónde se obtuvieron los datos? ¿Se están fusionando varias fuentes de datos? Si es así, ¿hay áreas que podrían plantear un problema al fusionar?

Los datos se sacaron de fatsecret, ya que es una calculadora nutrimental. Al principio había juntado algunas etiquetas y la calculadora, pero después consideré mejor solo usar la calculadora para que no afectará por cualquier cosa.

¿Hay algún plan para manejar los valores faltantes en cada una de las fuentes de datos?

Si, es buscar en otra página (preferentemente que sea verificada) y si no se encuentra nada, tomar el valor como 0.

¿Cuántos datos están accesibles o disponibles y cómo está la calidad de los mismos?

En lo personal he encontrado todos, no he tenido problema alguno, y en cuanto a calidad considero que están bastante bien, el único problema es que muchas veces debes hacer multiplicaciones si comiste más de una ración.

¿Cuál es la relación de los datos y la hipótesis del proyecto?

Que los datos recabados nos podrán ayudar a tomar decisiones respecto a nuestra alimentación, es decir, si vemos que no comemos mucho de un nutriente o nos faltan calorías, buscar algún especialista que nos pueda apoyar, y lo mismo pasa si tenemos un exceso.

https://colab.research.google.com/drive/1nec1Rjp7mhySGudmvivGSc2tLAjOvE8N?usp=sharing

Fase 3. Preparación de los datos

Parte 1: Selección, limpieza y preparación de los Datos en Python

```
print(X_train) # imprimimos x_train para comprobar que se realizó correctamente la preparación de los datos

print(y_train) # imprimimos y_train para comprobar que se realizó correctamente la preparación de los datos
```

```
[[5.330e+01 2.790e+01 6.000e+00 1.462e+03]
 [1.960e+00 1.521e+01 1.301e+01 2.110e+02]
 [2.390e+01 1.560e+01 3.100e+00 3.400e+01]
 [2.200e+01 2.000e+01 4.700e+01 3.500e+02]
 [2.970e+00 3.392e+01 2.953e+01 5.130e+02]
 [1.410e+00 2.424e+01 5.637e+01 1.779e+03]
 [1.410e+00 2.424e+01 5.637e+01 1.779e+03]
 [1.260e+01 1.200e+00 2.400e+00 2.420e+02]
 [0.000e+00 1.544e+01 5.910e+01 7.860e+02]
 [3.188e+01 3.202e+01 9.740e+00 4.040e+02]
 [6.670e+01 3.096e+01 4.138e+01 1.542e+03]
 [2.390e+01 1.560e+01 3.100e+00 3.400e+01]
 [2.365e+01 3.060e+01 2.495e+01 3.900e+02]
 [1.900e+01 2.000e+00 7.000e+00 1.470e+02]
 [0.000e+00 7.270e+00 1.088e+01 9.100e+01]
 [2.390e+01 1.560e+01 3.100e+00 3.400e+01]
 [2.700e+01 0.000e+00 1.000e+00 1.150e+02]
 [0.000e+00 7.720e+00 2.955e+01 3.930e+02]
 [5.500e+01 2.100e+01 1.800e+01 1.068e+03]
 [3.405e+01 1.725e+01 1.712e+01 2.440e+02]
 [2.200e+01 2.000e+01 4.700e+01 3.500e+02]
 [2.390e+01 1.560e+01 3.100e+00 3.400e+01]
 [2.390e+01 1.560e+01 3.100e+00 3.400e+01]
 [3.200e+01 1.900e+01 3.000e+00 2.200e+02]
 [2.390e+01 1.560e+01 3.100e+00 3.400e+01]
 [5.610e+01 9.000e-01 1.680e+00 6.000e+00]
 [3.546e+01 2.990e+01 3.317e+01 9.940e+02]
 [4.413e+01 1.420e+00 6.840e+00 1.500e+01]
 [2.000e+01 3.000e+00 2.000e+00 8.500e+01]
 [3.546e+01 2.990e+01 3.317e+01 9.940e+02]
 [5.500e+01 2.100e+01 1.800e+01 1.068e+03]
 [2.310e+01 2.100e+00 2.200e+00 1.820e+02]
 [2.220e+00 1.958e+01 2.134e+01 9.560e+02]
 [2.390e+01 1.560e+01 3.100e+00 3.400e+01]
 [0.000e+00 7.270e+00 1.088e+01 9.100e+01]
 [3.405e+01 1.725e+01 1.712e+01 2.440e+02]
```

```
dataset.columns # vemos los nombres de nuestras columnas para asignarlos a las variables
```

```
Index(['Calorías (kcal)', 'Carbohidratos (g)', 'Lípidos/grasas (g)',
       'Proteína (g)', 'Sodio (mg)'],
      dtype='object')
```

```
[24] X = dataset[['Carbohidratos (g)', 'Lípidos/grasas (g)', 'Proteína (g)', 'Sodio (mg)']].values # variables independientes

     y = dataset['Calorías (kcal)'].values # variable dependiente
```

```
[25] from sklearn.model_selection import train_test_split # importamos la herramienta para dividir los datos de SciKit-Learn

     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0) # asignación de los datos 80% p
```

```
print(X_train) # imprimimos x_train para comprobar que se realizó correctamente la preparación de los datos

print(y_train) # imprimimos y_train para comprobar que se realizó correctamente la preparación de los datos
```

```
[[5.330e+01 2.790e+01 6.000e+00 1.462e+03]
 [1.960e+00 1.521e+01 1.301e+01 2.110e+02]
 [2.390e+01 1.560e+01 3.100e+00 3.400e+01]
 [2.200e+01 2.000e+01 4.700e+01 3.500e+02]
 [2.970e+00 3.392e+01 2.953e+01 5.130e+02]
 [1.410e+00 2.424e+01 5.637e+01 1.779e+03]
 [1.410e+00 2.424e+01 5.637e+01 1.779e+03]
 [1.260e+01 1.200e+00 2.400e+00 2.420e+02]
 [0.000e+00 1.544e+01 5.910e+01 7.860e+02]
 [3.188e+01 3.202e+01 9.740e+00 4.040e+02]
 [6.670e+01 3.096e+01 4.138e+01 1.542e+03]
 [2.390e+01 1.560e+01 3.100e+00 3.400e+01]
 [2.365e+01 3.060e+01 2.495e+01 3.900e+02]
 [1.900e+01 2.000e+00 7.000e+00 1.470e+02]
 [0.000e+00 7.270e+00 1.088e+01 9.100e+01]
 [2.390e+01 1.560e+01 3.100e+00 3.400e+01]
 [2.700e+01 0.000e+00 1.000e+00 1.150e+02]
 [0.000e+00 7.720e+00 2.955e+01 3.930e+02]
```

✓ 0 s se ejecutó 11:19

Parte 2: Preparación de los datos

La selección de datos es fundamental para el desarrollo del proyecto, ya que implica el manejo de los datos, esto puede incluir la limpieza de datos, el cual es un análisis más detallado de los problemas en los datos que se han elegido incluir para el análisis.
También está involucrado el generar nuevos datos, ya que es frecuente que en ocasiones se necesite construir nuevos datos que aporten a nuestro análisis.
Por otra parte también se integran los datos, ya que al tener múltiples fuentes de datos para poder responder al mismo conjunto de preguntas, se podrían fusionar estos conjuntos de datos que contienen el mismo identificador único, para que al analizarlos se conecten y den un resultado más amplio.
Y por último es importante el formato de datos, ya que es útil verificar si ciertas técnicas requieren un formato u orden en particular para los datos, ya que es necesario ordenar los datos de alguna forma para después ejecutar el modelo. Incluso si el modelo puede realizar la clasificación de manera integrada, es posible ahorrar tiempo de procesamiento al hacer este paso antes de modelar.
Con esto en mente, la preparación de datos será más eficaz y obtendremos mejores resultados de nuestros datos registrados.

1. Responde las siguientes preguntas y justifica tu respuesta para cada una de las preguntas.

1. ¿Qué datos hay que seleccionar? Por qué.

En el caso del proyecto hicimos una selección de elementos, es decir, tomamos los datos con los que se cuenta, como son los carbohidratos, calorías, lípidos, proteínas y sodio. Esto es porque los datos en fila ya los tenemos y son los que queremos tomar en cuenta para nuestro análisis.

2. ¿Hay que eliminar o reemplazar valores en blanco? Sí / No / Por qué.

No, porque al realizar la limpieza no tuvimos ningún valor nulo y además al validar todos nuestros resultados dieron 0.

3. ¿Es posible agregar más datos? Sí / No / Por qué.

Si, pero se debería realizar el análisis de nuevo para comprobar que no haya ningún error, esto es porque estamos usando nuestra base de datos en excel, y si cambiamos el archivo donde existan más, habrá más datos que analizar.

4. ¿Hay qué integrar o fusionar datos de varias fuentes? Sí / No / Por qué.

Si, es recomendado usar fuentes fiables como la etiqueta o calculadoras de calorías, pero se debería evitar páginas no verificadas o que la información no sea muy confiable, esto es porque las calculadoras de calorías toman en cuenta el valor de las etiquetas, aunque suelen variar poco en algunas ocasiones.

5. ¿Es necesario ordenar los datos para el análisis? Sí / No / Por qué.

No, lo que si es necesario es tener el formato, es decir poner el valor de las calorías donde corresponda y el valor de los nutrientes donde corresponda dependiendo la fecha en el se consuma, pero ordenarlo no cambiará mucho pero si es recomendable porque puedes darte idea en que fechas fueron donde consumiste más calorías y además mantienes un orden y es más comprensible.

6. ¿Tengo que hacer conjuntos de datos para entrenamiento y prueba? Sí / No / Por qué.

Si, para dividir las variables en como queremos la información, en este caso utilizamos en la variable x el conjunto de datos de los nutrientes, y en y las calorías.

7. ¿Qué ajustes se tuvieron que hacer a los datos (agregar, integrar, modificar registros (filas), cambiar atributos (columnas)?

Se dividieron los datos scikit-learn para después hacer una asignación del 80% de los datos para entrenamiento y el 20% para pruebas.

https://colab.research.google.com/drive/1his2Ne6Po7vu4S7uyAOU39RzkVPKM38b?usp=sharing

Fase 4. Modelación de los datos

Parte 1: Análisis de regresión en Python

```
[4] import pandas as pd # importa la librería pandas y la asigna a la variable pd
    datosread = pd.read_excel('A01351746Registro-1.xlsx') # indicamos el nombre de nuestro archivo a ser leído
    datosread.head()
```

	Fecha (dd/mm/aa)	Momento	Nombre alimento	Calorías (kcal)	Carbohidratos (g)	Lípidos/grasas (g)	Proteína (g)	Sodio (mg)	Fuente
0	2021-02-14	desayuno	Consome de barbacoa	238	1.49	14.18	25.34	834	fatsecret
1	2021-02-14	comida	hamburguesa	552	35.46	29.90	33.17	994	fatsecret
2	2021-02-14	cena	hotdog	284	22.85	16.59	10.24	919	fatsecret
3	2021-02-15	desayuno	3 salchichas de pavo	465	1.41	24.24	56.37	1779	fatsecret
4	2021-02-15	comida	3 tostadas de pollo	426	29.52	21.15	30.33	606	fatsecret

```
[5] datosread.groupby("Momento").count() # con la función groupby agrupamos los datos de la columna Momento y con count() lo
```

Momento	Fecha (dd/mm/aa)	Nombre alimento	Calorías (kcal)	Carbohidratos (g)	Lípidos/grasas (g)	Proteína (g)	Sodio (mg)	Fuente
cena	22	22	22	22	22	22	22	22
comida	97	97	97	97	97	97	97	97
desayuno	97	97	97	97	97	97	97	97
snack	85	85	85	85	85	85	85	85

```
[6] datosread.describe()
```

	Calorías (kcal)	Carbohidratos (g)	Lípidos/grasas (g)	Proteína (g)	Sodio (mg)
count	301.000000	301.000000	301.000000	301.000000	301.000000
mean	325.172757	28.161595	15.650100	18.142691	615.491694
std	213.743983	22.207127	13.155141	18.175980	636.121333
min	12.000000	0.000000	0.000000	0.000000	0.000000
25%	156.000000	11.980000	4.620000	2.400000	115.000000
50%	280.000000	24.000000	15.210000	10.880000	350.000000
75%	465.000000	44.130000	24.240000	29.530000	994.000000
max	1192.000000	92.000000	70.360000	63.480000	2757.000000

```
[7]
datosseleccion = datosread.iloc[:,3:8] # : selecciona todas las filas y 3:8(-1) seleccion columnas de la 4 la 7

datosseleccion # desplegamos el dataframe
```

	Calorías (kcal)	Carbohidratos (g)	Lípidos/grasas (g)	Proteína (g)	Sodio (mg)
0	238	1.49	14.18	25.34	834
1	552	35.46	29.90	33.17	994
2	284	22.85	16.59	10.24	919
3	465	1.41	24.24	56.37	1779
4	426	29.52	21.15	30.33	606
...	...	...	...	...	...
296	705	82.83	26.17	33.53	1399
297	119	19.00	2.00	7.00	147
298	704	66.70	30.96	41.38	1542
299	12	1.51	0.29	0.95	792
300	115	20.00	3.00	2.00	85

301 rows × 5 columns

```
[8] datosseleccion.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301 entries, 0 to 300
Data columns (total 5 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Calorías (kcal)     301 non-null    int64
 1   Carbohidratos (g)   301 non-null    float64
 2   Lípidos/grasas (g)  301 non-null    float64
 3   Proteína (g)        301 non-null    float64
 4   Sodio (mg)          301 non-null    int64
dtypes: float64(3), int64(2)
memory usage: 11.9 KB
```

```
[9] datosseleccion.isnull().values.any() # buscamos valores nulos y obtenemos True o False dependiendo si hay o no

False
```

```
[11] dataset = datosseleccion.dropna() # creamos un nuevo dataframe descartando los valores nulos o vacíos de nuestro datafra
```

```
[12]
dataset.isnull().sum() # validamos que no tenemos valores nulos en ninguna columna, todos deben dar cero
```

```
Calorías (kcal)       0
Carbohidratos (g)     0
Lípidos/grasas (g)    0
Proteína (g)          0
Sodio (mg)            0
dtype: int64
```

```
[13] dataset.columns # vemos los nombres de nuestras columnas para asignarlos a las variables

     Index(['Calorías (kcal)', 'Carbohidratos (g)', 'Lípidos/grasas (g)',
            'Proteína (g)', 'Sodio (mg)'],
           dtype='object')
```

```
[15] X = dataset[['Carbohidratos (g)', 'Lípidos/grasas (g)', 'Proteína (g)', 'Sodio (mg)']].values # variables independientes

     y = dataset['Calorías (kcal)'].values # variable dependiente
```

```
[16] n sklearn.model_selection import train_test_split # importamos la herramienta para dividir los datos de SciKit-Learn

     rain, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0) # asignación de los datos 80% para
```

```
[17] from sklearn.linear_model import LinearRegression # importamos la clase de regresión lineal

     modelo_regresion = LinearRegression() # modelo de regresión
```

```
[18] modelo_regresion.fit(X_train, y_train) # aprendizaje automático con base en nuestros datos

     LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
[19] x_columns = ['Carbohidratos (g)', 'Lípidos/grasas (g)', 'Proteína (g)', 'Sodio (mg)']
     coeff_df = pd.DataFrame(modelo_regresion.coef_, x_columns, columns=['Coeficientes'])
     coeff_df # despliega los coefientes y sus valores; por cada unidad del coeficente, su impacto en las calorías será igual
```

	Coeficientes
Carbohidratos (g)	3.786037
Lípidos/grasas (g)	9.021360
Proteína (g)	4.220219
Sodio (mg)	-0.000155

```
[20] y_pred = modelo_regresion.predict(X_test) # probamos nuestro modelo con los valores de prueba
```

```
[21] validacion = pd.DataFrame({'Actual': y_test, 'Predicción': y_pred, 'Diferencia': y_test-y_pred}) # creamos un dataframe

     muestra_validacion = validacion.head(25) # elegimos una muestra con 25 valores

     muestra_validacion # desplegamos esos 25 valores
```

	Actual	Predicción	Diferencia
0	72	72.695389	-0.695389
1	110	113.633770	-3.633770
2	72	72.695389	-0.695389
3	705	692.082133	12.917867
4	417	416.209158	0.790842
5	115	112.321047	2.678953
6	115	112.321047	2.678953
7	552	544.930836	7.069164
8	136	158.287618	-22.287618
9	71	69.729635	1.270365
10	220	209.861100	10.138900
11	142	139.376587	2.623413
12	248	245.405885	2.594115
13	333	336.146599	-3.146599
14	156	158.669694	-2.669694
15	852	838.851942	13.148058
16	115	112.321047	2.678953

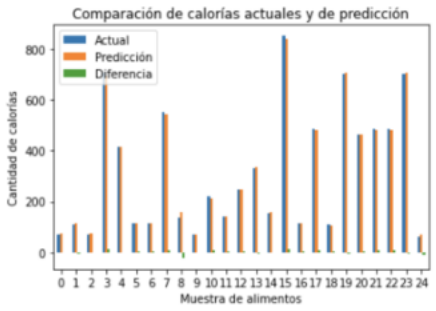
```
[21]
```

17	488	479.695007	8.304993
18	112	107.534350	4.465650
19	704	707.332182	-3.332182
20	465	462.742564	2.257436
21	488	479.695007	8.304993
22	488	479.695007	8.304993
23	704	707.332182	-3.332182
24	62	69.796421	-7.796421

```
[22] validacion["Diferencia"].describe()

count    61.000000
mean     -1.050083
std      11.916650
min     -50.060405
25%      -2.854265
50%       2.594115
75%       4.465650
max      13.148058
Name: Diferencia, dtype: float64
```

```
[23] from sklearn.metrics import r2_score # importamos la métrica R cuadrada (coeficiente de determinación)

     r2_score(y_test, y_pred) # ingresamos nuestros valores reales y calculados

     0.9977799180275979
```

```
[24] import matplotlib.pyplot as plt # importamos la librería que nos permitirá graficar

     muestra_validacion.plot.bar(rot=0)  # creamos un gráfico de barras con el dataframe que contiene nuestros datos actuales

     plt.title("Comparación de calorías actuales y de predicción") # indicamos el título del gráfico

     plt.xlabel("Muestra de alimentos") # indicamos la etiqueta del eje de las x, los alimentos

     plt.ylabel("Cantidad de calorías") # indicamos la etiqueta del eje de las y, la cantidad de calorías

     plt.show() # desplegamos el gráfico
```



Parte 2: Modelación de los datos

Describe con tus palabras en qué consiste la Fase 4: Modelación de los datos

Durante esta fase, el modelado se realiza generalmente en múltiples iteraciones. Normalmente, se ejecutan varios modelos utilizando parámetros predeterminados y luego se deben ajustar dichos parámetros para las manipulaciones requeridas por el modelo que se eligió.

También se debe tomar en consideración la modelación que se quiere realizar, ya que con estos será más fácil exponer los datos y entenderlos. Para escoger el

modelo que se quiera usar, se recomienda tener en cuenta los tipos de datos disponibles, los objetivos planteados y algo especifico que requiera.

Al examinar los resultados de un modelo, es necesario asegurarse de tomar notas sobre cada experiencia del modelado. Esto implica el registrar y guardar notas de cada modelo utilizado. Y lo último sería evaluar el modelo referente a lo anteriormente mencionado.

¿Cuántos intentos o corridas realizaste para obtener los resultados sin errores? Porqué
En mi caso no tuve ningún error al realizar esta práctica, pero en las anteriores al dejar mis datos habían casillas nombradas como NA, pero actualmente ese problema ya no existe.
¿Cómo los resolviste los problemas que se presentaron?
Para realizar las actividades tomo en cuenta las instrucciones de canvas, al igual de los correos y recomendaciones en clases.

¿Qué resultados arrojó el análisis? Incluye imagen de cada resultado y explica cada uno de los resultados:

Estadística descriptiva

	Calorías (kcal)	Carbohidratos (g)	Lípidos/grasas (g)	Proteína (g)	Sodio (mg)
count	301.000000	301.000000	301.000000	301.000000	301.000000
mean	325.172757	28.161595	15.650100	18.142691	615.491694
std	213.743983	22.207127	13.155141	18.175980	636.121333
min	12.000000	0.000000	0.000000	0.000000	0.000000
25%	156.000000	11.980000	4.620000	2.400000	115.000000
50%	280.000000	24.000000	15.210000	10.880000	350.000000
75%	465.000000	44.130000	24.240000	29.530000	994.000000
max	1192.000000	92.000000	70.360000	63.480000	2757.000000

Coeficientes de regresión
Los coeficientes de cada variable; en este caso notamos como el Sodio tiene un impacto no significativo en la cantidad de calorías al tener un coeficiente sumamente pequeño y negativo.

	Coeficientes
Carbohidratos (g)	3.786037
Lípidos/grasas (g)	9.021360
Proteína (g)	4.220219
Sodio (mg)	-0.000155

Valores actuales y de predicción

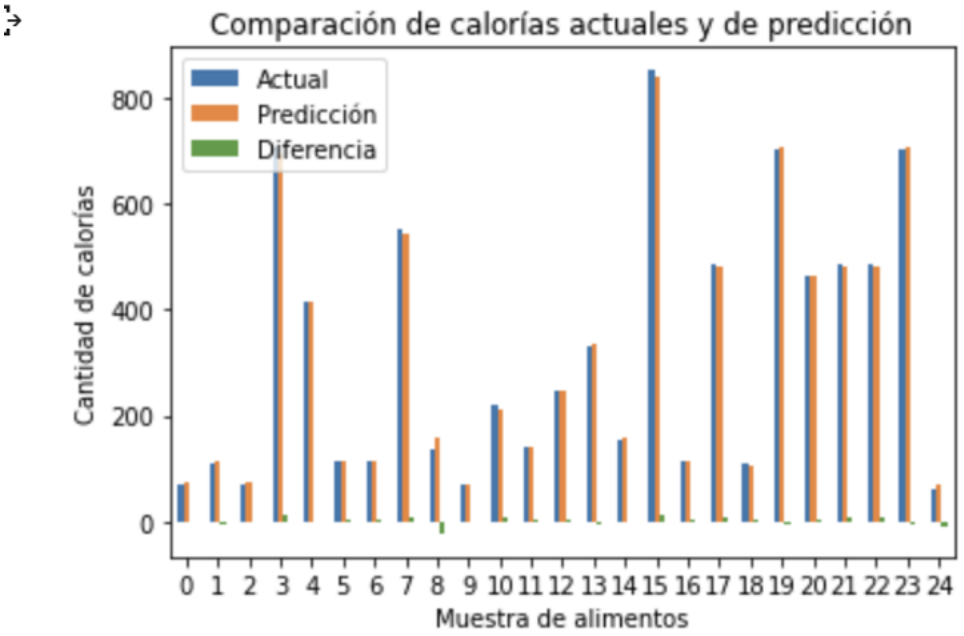
```
count    61.000000
mean     -1.050083
std      11.916650
min     -50.060405
25%      -2.854265
50%       2.594115
75%       4.465650
max      13.148058
Name: Diferencia, dtype: float64
```

Coeficiente de determinación r2

En este caso el valor de R2 es de 0.9978 por lo que podemos concluir que el modelo explica el 99% del contenido calórico de los alimentos.

```
0.9977799180275979
```

Gráfica



¿Cuáles son tus conclusiones de la modelación?

Se observa una predicción bastante acertada al no haber mucha diferencia entre los valores actuales de las calorías y los valores calculados, por lo que podría considerar que la fase 4 fue un éxito.

https://colab.research.google.com/drive/1his2Ne6Po7vu4S7uyAOU39RzkVPKM38b?usp=sharing

Efecto del consumo calórico en el tiempo

```
[12] import pandas as pd # importa la librería pandas y la asigna a la variable pd
     datos_consumo = pd.read_excel('A01351746Registro.xlsx') # indicamos el nombre de nuestro archivo a ser leído
     datos_consumo.head()
```

	Fecha (dd/mm/aa)	Momento	Nombre alimento	Calorías (kcal)	Carbohidratos (g)	Lípidos/grasas (g)	Proteína (g)	Sodio (mg)	Fuente
0	2021-02-14	desayuno	Consome de barbacoa	238	1.49	14.18	25.34	834	fatsecret
1	2021-02-14	comida	hamburguesa	552	35.46	29.90	33.17	994	fatsecret
2	2021-02-14	cena	hotdog	284	22.85	16.59	10.24	919	fatsecret
3	2021-02-15	desayuno	3 salchichas de pavo	465	1.41	24.24	56.37	1779	fatsecret
4	2021-02-15	comida	3 tostadas de pollo	426	29.52	21.15	30.33	606	fatsecret

```
[13] datos = datos_consumo[["Fecha (dd/mm/aa)","Calorías (kcal)"]] # seleccionamos las dos columnas que necesitaremos
     datos.head() # imprimiendo los datos seleccionados
```

	Fecha (dd/mm/aa)	Calorías (kcal)
0	2021-02-14	238
1	2021-02-14	552
2	2021-02-14	284
3	2021-02-15	465
4	2021-02-15	426

```
[14]
     suma_calorias = datos["Calorías (kcal)"].sum()
     suma_calorias # despliega el total de calorias

     97877

[15] dias = datos["Fecha (dd/mm/aa)"].nunique()
     dias # despliega el total de días unicos

     97

[16] calorias_promedio = suma_calorias/dias # total de calorías consumidas entre el número de días que tomó consumirlas
     print("Tu promedio de calorías consumidas en", dias,"días es:", calorias_promedio)

     Tu promedio de calorías consumidas en 97 días es: 1009.0412371134021

[24]
     peso = int(input("Ingresa tu peso en kilogramos: "))

     altura = int(input("Ingresa tu altura en centimetros: "))

     edad = int(input("Ingresa tu edad en años: "))

     genero = input("Ingresa tu género, Mujer/Hombre: ")

     Ingresa tu peso en kilogramos: 82
     Ingresa tu altura en centimetros: 175
     Ingresa tu edad en años: 19
     Ingresa tu género, Mujer/Hombre: Hombre

[25] if(genero == "Mujer"):
        calorías_requeridas = 655+(9.56*peso)+(1.85*altura)-(4.68*edad) # fórmula para estimar calorías requeridas en mujer

     elif(genero == "Hombre"):
       calorías_requeridas = 66.5+(13.75*peso)+(5*altura)-(6.8*edad) # fórmula para estimar calorías requeridas en hombre

     print("Con base en tus datos, tu consumo de calorías al día debe ser de:", calorías_requeridas)

     Con base en tus datos, tu consumo de calorías al día debe ser de: 1939.8

[26] diferencia = calorias_promedio - calorías_requeridas

     diferencia

     -930.7587628865979

     diferencia * 450/3500 * 365 /1000 # realiza la proporción, se multiplica por 365 (días) y se divide entre 1000 (gramos)

     inuas con el consumo calórico actual, en un año tu cambio de masa corporal sería aproximadamente de:",efecto_anual,"kg")

     Si continuas con el consumo calórico actual, en un año tu cambio de masa corporal sería aproximadamente de: -43.679179086
```

https://colab.research.google.com/drive/1YAJT8veBLY9VQDXEQh553K4hYsw16ACt?usp=sharing

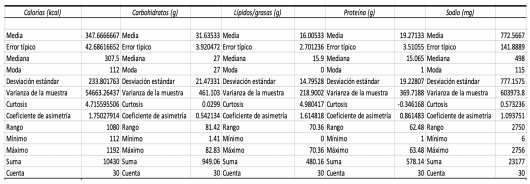
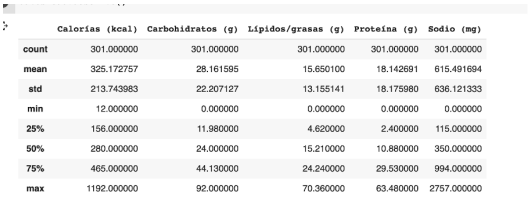
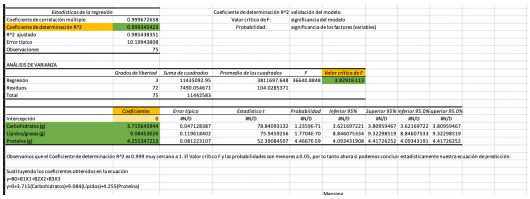
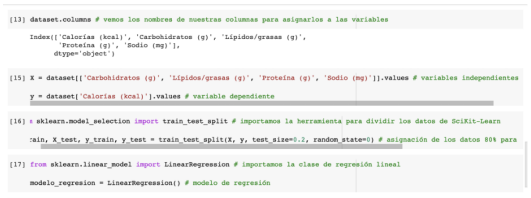
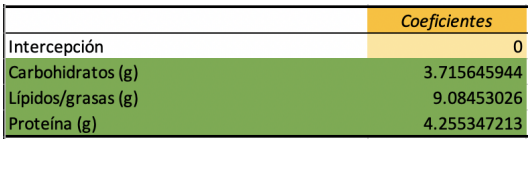
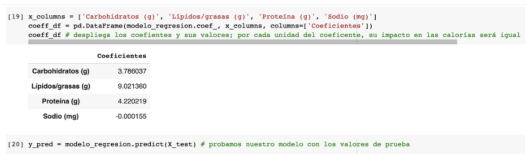
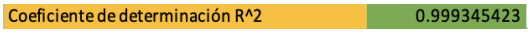
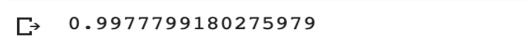
Reflexión final (Conclusiones)

1. Responde la hipótesis inicial: ¿Si consumo cierta cantidad calórica, puedo tener cambios en mi masa corporal (peso) en un determinado tiempo? De acuerdo a tus resultados en la estimación se acepta o se rechaza.

De acuerdo al efecto del consumo calórico en el tiempo, se puede observar que en un año mi masa corporal disminuirá, por lo que se acepta, pero cabe aclarar que al no tener en cuentas otras variables, el resultado puede verse afectado,

como el hecho de no ingresar bebidas a la base de datos o el ejercicio, entre otras variables, pero lo que si es un hecho, es que dependiendo el consumo de calorías de una persona, hará que disminuya o aumente de peso.

2. Compara los procedimientos y resultados de regresión realizados en Excel en la semana 4 y en Python en la semana 14. Realiza una tabla comparativa para explicar las diferencias, incluye imagen y explicación de cada resultado en Excel y Python. ¿Cuál te pareció mejor, por qué?

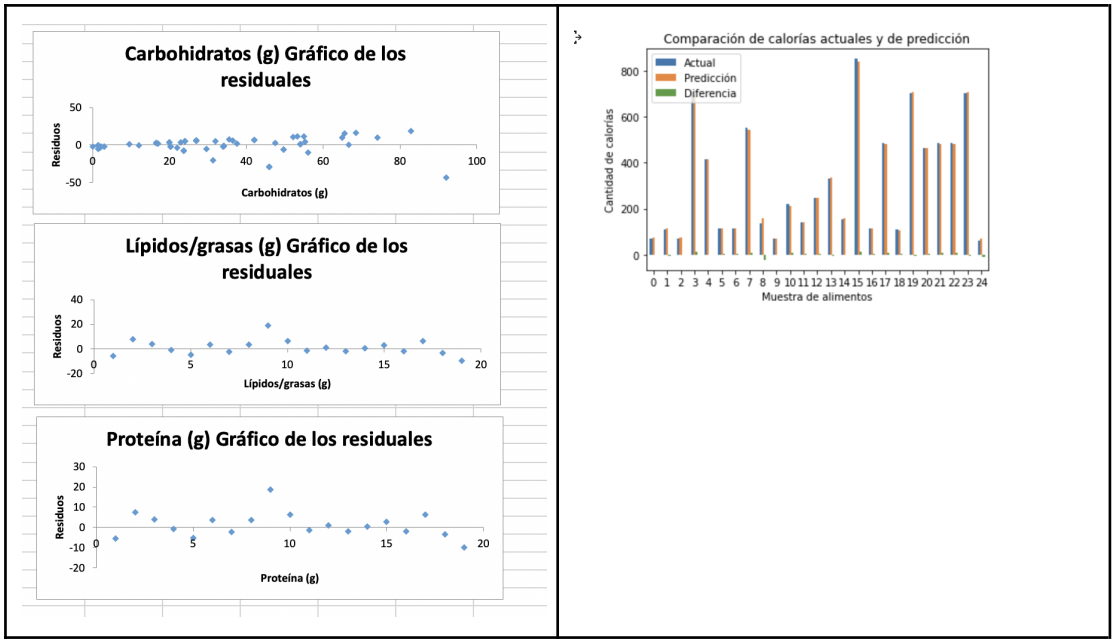
Excel	Python
	
	
	
	

	Coeficientes
Intercepción	0
Carbohidratos (g)	3.715645944
Lípidos/grasas (g)	9.08453026
Proteína (g)	4.255347213

Coeficiente de determinación R^2	0.999345423

	Coeficientes
Carbohidratos (g)	3.786037
Lípidos/grasas (g)	9.021360
Proteína (g)	4.220219
Sodio (mg)	-0.000155

0.9977799180275979



El análisis descriptivo de excel fueron durante las primeras semanas, por lo que habían menos datos, y al hacer el análisis en python se puede ver un aumento en la mayoría de las medias, por otra parte en los coeficiente de determinación obtuvimos un valor en el excel del 0.9993, mientras que en el python disminuyó a 0.9977.

En el caso del excel, la variable de sodio no se ocupa, es decir se quedó como calorías =0+3.715(Carbohidratos)+9.084(Lípidos)+4.255(Proteína), mientras que en el python la variable “y” son las calorías, mientras que las variables “x” son las proteínas, carbohidratos, lípidos y sodio.

Como se puede observar, las gráficas del excel son de los residuos de las variables utilizadas y mientras que en la de python genera una gráfica de barras de la comparación de calorías actuales y la predicción.

En conclusión este proyecto me ayudó a tomar una decisión sobre que hacer con mi alimentación, ya que gracias a la pandemia mi alimentación no fue la mejor, además que no hacía ejercicio o alguna actividad física, por lo que gracias a esto puedo darme una idea en como ir mejorando mi alimentación para no tener un exceso de calorías, por lo que seguiré con mi base de datos pero ahora agregando las bebidas y no solo los alimentos para obtener un resultado aún más real.

Referencias

Anónimo. (2020). Nueve industrias que aplican la Ciencia de Datos para solucionar problemas reales. 27 de marzo del 2021, de xataka Sitio web: https://www.xataka.com/n/nueve-industrias-que-aplican-ciencia-datos-para-solucionar-problemas-reales#:~:text=El%20mantenimiento%20predictivo%20es%20un,%2C%20nivel%20de%20ruido%2C%20etc.

Anónimo. (SF). Análisis de Datos. 27 de marzo del 2021, de question pro Sitio web: https://www.questionpro.com/es/analisis-de-datos.html#:~:text=El%20an%C3%A1lisis%20de%20datos%20se,hip%C3%B3tesis%20es%20cierta%20o%20no.

Gaby Juarez. (2017). Ciencia de Datos vs Analítica de Datos -¿Por qué es importante ?. 27 de marzo del 2021, de nexolution Sitio web: http://www.nexolution.com/ciencia-de-datos-vs-analitica-de-datos-porque-es-importante/

UTEC. (2021). ¿Qué es la Ciencia de Datos y para qué se utiliza?. 30 de mayo del 2021, de UTEC Sitio web: https://www.utec.edu.pe/blog-de-carreras/ciencias-de-datos/que-es-la-ciencia-de-datos-y-para-que-se-utiliza