



# Tecnológico de Monterrey

**Campus Sonora Norte**

**Nombre del trabajo:**

Evidencia 2. Proyecto de Ciencia de Datos

**Curso:**

Matemáticas y ciencia de datos para la toma de decisiones (Gpo 601)

**Alumnos:**

Hector Alán Gutiérrez Gálvez

**Matrícula:**

A01253031

**Profesor:**

Eduardo Antonio Hinojosa Palafox

### Introducción

La ciencia de datos es el estudio de datos con el fin de extraer información significativa para empresas. Tiene un enfoque multidisciplinario que combina prácticas dentro del campo de las matemáticas, la estadística, inteligencia artificial y la ingeniería de computación para analizar grandes cantidades de datos. Esto le permite a los científicos de datos poder realizar un análisis y poder tener respuesta de que es lo que está pasando. (Amazon, 2022)

Esta disciplina es muy importante porque combina herramientas, métodos y tecnología para encontrar patrones a partir de los datos. Actualmente estamos inundados de datos y entre más avanzamos los datos irán creciendo. Existen dispositivos que son capaces de recopilar y almacenar información de manera automática. En la red existen grandes cantidades de datos de texto, audio, vídeo e imágenes. (Amazon, 2022)

La Ciencia de Datos es una disciplina cada día cobra mucha popularidad e interés en el mundo. En el 2012 Forbes realizó una declaración en donde indicaba que la Ciencia de Datos sería la profesión más sexy del siglo 21.

Los beneficios que tiene la ciencia de datos para las empresas es que gracias a esto pueden descubrir patrones desconocidos de transformación, innovar con nuevos productos y soluciones y optimización en tiempo real. Muchas empresas independiente de su tamaño necesitan esto para impulsar el crecimiento y mantener una ventaja competitiva.(Amazon, 2022)

La intención de este proyecto es demostrar mediante la ciencia de datos conocer como esta mi dieta día a día. Esto para poder realizar un análisis que me muestre que es lo que debo de cambiar de mi dieta para poder tener una vida más saludable, en la cual se adapte a mis cualidades, algo que es muy importante considerar, ya que las dietas dependen mucho del peso de la persona entre otras cosas más.

## Fase 1. Entendimiento del negocio

El entendimiento del negocio trata sobre identificar los objetivos del negocio, ya que sin esto no se tiene una meta en la cual se quiere. Después se necesita evaluar la situación en la cual nos encontramos, para saber que tan adelante o atrás estamos para avanzar. Si se establecen bien los objetivos y se sabe cual es en verdad la situación se puede definir exactamente los objetivos precisos para que con la ciencia de datos se puedan lograr con eficacia.

Por consiguiente, con toda la información acumulada e investigada sobre los objetivos y la situación actual del negocio puedes desarrollar tu plan de trabajo, esto es muy importante porque será el pan de cada día en tu negocio y se realizará obviamente con base los objetivos para ser el trayecto para llegar a estos porque ya se tienen los conocimientos necesarios. El análisis de datos, por lo general se utiliza el contexto de la inteligencia de negocios. Por lo general las herramientas de análisis de datos son el último paso en la cadena de recopilación, estructuración y procesamiento de datos. Poniendo como ejemplo el proyecto que se va realizar, el objetivo es cómo afecta la alimentación que yo consumo y la información la recolectamos de lo que comes día a día con sus debidas calorías y nutrientes.

### 1. ¿Quién es el cliente?

Los clientes pueden ser empresas o personas que busquen resolver o identificar un problema.

### 2. ¿Qué problemas estás tratando de resolver?

Ayuda a las empresas reducir costos, aumentar la productividad y proporcionar datos para la toma de decisiones.

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

Las soluciones que tratará de proveer la ciencia de datos es exponer tendencias y producir información que las empresas pueden usar para tomar mejores decisiones, crear productos y servicios innovadores. En el caso del

aumento de propina se enfocaron más en los lugares donde reciben más propina e hicieron un análisis.

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

Se necesita entender el problema que se quiere resolver. Identificar alguna meta objetivo que se desee lograr, es importante entender que necesitarás conocer para lograr esto.

5. ¿Qué deberás hacer para desarrollar tu solución?

Lo que se debe hacer para desarrollar tu solución es entender los datos que se tiene y analizarlos e innovar con una solución que combata el problema y hacerlo más eficiente.

## **Fase 2. Entendimiento de los datos**

En resumidas cuentas, lo que nos dice es que la recolección de datos es una parte esencial de la ciencia de datos, ya que esto es lo que le da valor al análisis que se quiere realizar, tenemos 3 tipos de datos, existentes, que son los datos que se tienen en este caso por ejemplo mi peso, edad y altura, luego le siguen los datos adquiridos, que son los que se encuentran en Excel, los carbohidratos, proteínas y así. Por último los datos adicionales son los que como el nombre lo dice son adicionales y en algún momento nos podrían ayudar, como lo pueden ser encuestas, revistas, etc.

Hay casos en los cuales no se pueden analizar bien los datos porque se producen errores, los errores más comunes son los que están en blanco o que no tienen una respuesta concreta, errores tipográficos, errores de medición como lo puede ser que se ingresa cm en algo que pide Celsius y así hay muchos de los errores que se pueden cometer. Entre más datos se obtiene más preciso el modelo puede llegar a ser, dentro de estos datos existen de tipo numérico, categórico o booleano. Después

sigue analizar los datos y se realiza una hipótesis que ayude a dar forma en la transformación de los datos.

La presentación de los datos en el cual se visualice en realidad cual es el problema conducirá a decisiones más claras y concisas. Los gráficos circulares o de dona sirven mucho para las comparaciones parciales, el diagrama de Sankey es una forma muy visual de poder determinar el flujo de volumen de información.

1. ¿Cuáles son tus datos existentes (registrados), datos adquiridos (datos externos) y datos adicionales (datos generados)?

Mis datos existentes son los datos que se tienen en este caso por ejemplo mi peso, edad y altura. Los datos adquiridos son los que se encuentran en Excel, los carbohidratos, proteínas entre otros. Y por último los datos adicionales son los que como el nombre lo dice son adicionales y en algún momento nos podrían ayudar, como lo pueden ser encuestas, revistas, etc.

2. ¿Qué tipos de datos se analizarán?

Los tipos de datos que se analizaron son de tipo continuo de proporción, ya que los datos son numéricos como por ejemplo la altura y el peso. Además, un valor cero que es la ausencia de la característica y la razón entre dos valores resulta significativa.

3. ¿Qué atributos (columnas) de la base de datos parecen más prometedores?

Yo considero que los datos que son más prometedores son los nutrientes, ya que de ahí fue donde se realizó el análisis. Los nutrientes son los datos que más peso tiene sin dejar afuera a las calorías que también son importantes, pero a mi parecer los nutrientes son los datos más valiosos por el hecho de que en base a eso nos puede mostrar una respuesta más

detallada, ya que con la ayuda de los nutrientes se puede sacar las calorías, como se había realizado en un ejercicio pasado.

4. ¿Qué atributos parecen irrelevantes y pueden ser excluidos?

Los atributos que son irrelevantes podrían ser la fuente de donde se realizó la extracción de los datos, ya que de ahí se hizo toda la extracción.

5. ¿Hay datos suficientes (filas) para sacar conclusiones generalizables o hacer predicciones precisas?

Ya que se tienen muy pocos datos, no creo que se puedan sacar conclusiones generalizables o predicciones precisas porque lo recomendable fue obtener 300 filas, por lo cual me quede muy corto.

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

No creo que existan demasiados atributos para poder realizar un modelo fácil de interpretar. Los atributos son fáciles de interpretar, ya que son claros y concisos, a lo mejor la fecha podría ser algo difícil de interpretar, pero fuera de ahí ya no.

7. ¿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 obtuvieron de la aplicación myfitnesspal, que es una aplicación que la utiliza mucha gente que tiene una dieta estricta y necesita tener un registro de los nutrientes que ingiere cada día. Ya que solo se utilizó esa fuente no se combinó con otra.

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

El plan que se podría ejecutar en caso de que haya un valor faltante podría ser asignándole 0, ya que la mayoría de los valores son numéricos

y también existen alimentos que en veces no contiene ciertos tipos de nutrientes como el agua.

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

Gracias a la función shape se tienen 2,288 datos disponibles. Los datos accesibles vienen siendo la fecha, momento, nombre del alimento, calorías, carbohidratos, lípidos/grasas, proteína y sodio. Dentro de las filas en cada columna se pueden ver el número de nutrientes que contiene cada alimento.

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

La relación que existe entre los datos y la hipótesis del proyecto es que en base los datos que se obtuvieron se podría de alguna manera comprobar si estoy ingiriendo los nutrientes suficientes con base a los datos existentes que se tienen.

### **Fase 3. Preparación de los datos**

La fase 3 consiste en la preparación de datos, que se estima que es la parte más importante, ya que toma del 50 al 70% del tiempo y el esfuerzo del proyecto. Este mismo involucra las siguientes tareas: Selección de datos, Limpieza de datos, Generación de nuevos datos, Integración de datos, Formato de los datos y por último División de datos en conjuntos para entrenamiento y prueba.

Dentro de lo que es selección de datos es básicamente seleccionar los datos relevantes que se van a analizar, existen dos tipos de seleccionar datos. Después seguimos con la limpieza de datos que aquí implica un análisis más detallado de los

problemas en los datos que se han elegido. La generación de nuevos datos se utiliza en casos que se tenga que crear una nueva columna que nos ayude a entender más los datos. La integración de datos es fusionar los datos que tengan diferentes múltiples fuentes para que puedan ser más fácil analizarlos y tener un resultado más amplio, hay 2 métodos básicos de integración de datos. Ya para terminar el formato de datos viene siendo ordenar los datos de alguna forma para después ejecutar el modelo, esto para poder ahorrar tiempo de procesamiento antes de modelar.

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

Los datos que se tienen que seleccionar son los nutrientes (calorías, carbohidratos, lípidos/ grasas, proteínas y sodio), que son 5 columnas y 391 filas en total. Se seleccionaron estos porque son los datos más relevantes y son los que más nos brindan información.

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

No, gracias a las siguientes funciones:

- `datos seleccionados.isnull().values.any()`
- `dataset = datos_seleccionados.dropna()`
- `dataset.isnull().sum()`

Que fueron utilizadas en Google Colab pude observar que no existe ningún valor nulo, ya que en la primera función no me mostro ningún True demostrando de que si exista un valor nulo. Después se creó un nuevo data frame con los mismos datos seleccionados y volvió a validar si había valores nulos y todos dieron 0, por lo cual no existe ninguno.

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



Si, yo considero que podría ser útil crear una nueva columna en la cual se cuentan las calorías que se quemaron en el día, ya que por lo general no siempre contamos con las calorías con las que ingerimos porque por lo general tendemos a hacer actividades físicas en las cuales perdemos calorías y nutrientes. Esto nos podría ayudar a tener un marco más claro del análisis que se está realizando.

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

No, porque todos los datos fueron extraídos de la misma fuente, que es Myfittnespal. Gracias a esto no es necesario fusionarlos, por lo cual se puede analizar de una mejor manera y se puede tener un resultado mucho más amplio. Esto fue considerado antes de realizar el análisis, ya que si se tuvo en cuenta de que la integración de datos puede volverse algo complejo si no se le dedica un buen tiempo para poder comprender muy bien los datos. En esta etapa se podría considerar que los datos más relevantes son los nutrientes, pero si nos ponemos más específicos, son las calorías, ya que con esto yo considero que podría ser más fácil poder contestar preguntas o hipótesis del proyecto que se está realizando.

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

Yo considero que no es necesario utilizar técnicas para realizar un formato u orden particular para los datos, porque el algoritmo que se tiene no considera que se requiera que los datos estén clasificados antes o después de ejecutar el modelo. Cabe recalcar que si se clasifican los datos me podría ayudar a ahorrar tiempo de procesamiento al hacer este paso antes de modelar, pero como se tienen pocos datos, yo considero que no es muy necesario este paso.

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

Si, se asignaron 80% de los datos para entrenamiento y 20% para prueba, esto porque con los datos con los que partimos pueda entrenar a mi algoritmo, es decir, las características y las etiquetas. Mientras que por la parte de prueba, los datos que se seleccionaron se pueda comprobar si el modelo que se ha generado a partir de los datos de entrenamiento funciona o también si las respuestas predichas por el modelo para un caso totalmente nuevo son acertadas o no.

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

No se realizó ningún ajuste a los datos, más que eliminar columnas, que fueron la de la fecha, momento, nombre del alimento y la fuente, por el momento solo se dejaron los datos de los nutrientes, ya que como se había mencionado anteriormente, son los que yo considero que son los datos más relevantes a mi parecer.

## Fase 4. Modelación de los datos

### Parte 1: Análisis de regresión en Python

```
import pandas as pd # importa la libreria pandas y la asigna a la variable pd

[ ] datos_consumo = pd.read_excel('A01253031_Registro-1_Avance.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 | 2022-08-17       | Desayuno | Huevo con tortilla       | 303             | 23.0              | 17.0               | 16.0         | 233        | MyFitnessPal |
| 1 | 2022-08-17       | Comida   | Pollo asado              | 422             | 2.0               | 19.0               | 60.0         | 2          | MyFitnessPal |
| 2 | 2022-08-17       | Cena     | Quesadillas con frijoles | 572             | 51.0              | 29.0               | 29.0         | 702        | MyFitnessPal |
| 3 | 2022-08-17       | Snack    | Nito                     | 252             | 37.0              | 10.0               | 3.0          | 10         | MyFitnessPal |
| 4 | 2022-08-17       | Snack    | 5 Picafrasas             | 100             | 25.0              | 0.0                | 0.0          | 225        | MyFitnessPal |

```
datos_consumo.groupby("Momento").count() # con la función groupby agrupamos los datos de la columna Momento y con count() los contamos para obtener subtotales
```

|          | Fecha (dd/mm/aa) | Nombre alimento | Calorías (kcal) | Carbohidratos (g) | Lípidos/grasas (g) | Proteína (g) | Sodio (mg) | Fuente |
|----------|------------------|-----------------|-----------------|-------------------|--------------------|--------------|------------|--------|
| Momento  |                  |                 |                 |                   |                    |              |            |        |
| Cena     | 78               | 78              | 78              | 78                | 78                 | 78           | 78         | 78     |
| Comida   | 78               | 78              | 78              | 78                | 78                 | 78           | 78         | 78     |
| Desayuno | 78               | 78              | 78              | 78                | 78                 | 78           | 78         | 78     |
| Desayuno | 2                | 2               | 2               | 2                 | 2                  | 2            | 2          | 2      |
| Snack    | 157              | 157             | 157             | 157               | 157                | 157          | 157        | 157    |

```
[ ] datos_consumo.describe()
```

## Evidencia 2. Proyecto de Ciencia de Datos

```
[ ]
```

|       | Calorías (kcal) | Carbohidratos (g) | Lípidos/grasas (g) | Proteína (g) | Sodio (mg)  |
|-------|-----------------|-------------------|--------------------|--------------|-------------|
| count | 391.000000      | 391.000000        | 391.000000         | 391.000000   | 391.000000  |
| mean  | 327.138107      | 31.956206         | 15.460102          | 16.352174    | 409.051151  |
| std   | 206.084178      | 25.044617         | 11.659966          | 14.591886    | 523.009822  |
| min   | 40.000000       | 0.000000          | 0.000000           | 0.000000     | 0.000000    |
| 25%   | 165.000000      | 16.000000         | 7.000000           | 2.000000     | 180.000000  |
| 50%   | 300.000000      | 30.000000         | 13.000000          | 12.000000    | 320.000000  |
| 75%   | 406.000000      | 37.200000         | 21.950000          | 29.000000    | 702.000000  |
| max   | 1061.000000     | 115.000000        | 68.000000          | 65.000000    | 3110.000000 |

```

datos_seleccionados = datos_consumo.iloc[:,3:8] # : selecciona todas las filas y 3:8(-1) seleccion columnas de la 4 la 7
datos_seleccionados # desplegamos el dataframe

```

```
[ ]
```

|     | Calorías (kcal) | Carbohidratos (g) | Lípidos/grasas (g) | Proteína (g) | Sodio (mg) |
|-----|-----------------|-------------------|--------------------|--------------|------------|
| 0   | 303             | 23.0              | 17.0               | 16.0         | 233        |
| 1   | 422             | 2.0               | 19.0               | 60.0         | 2          |
| 2   | 572             | 51.0              | 29.0               | 29.0         | 702        |
| 3   | 252             | 37.0              | 10.0               | 3.0          | 10         |
| 4   | 100             | 25.0              | 0.0                | 0.0          | 225        |
| ... | ...             | ...               | ...                | ...          | ...        |
| 386 | 366             | 26.3              | 26.5               | 8.0          | 334        |
| 387 | 330             | 30.0              | 22.0               | 3.0          | 500        |
| 388 | 1052            | 115.0             | 49.0               | 36.0         | 1710       |
| 389 | 470             | 52.0              | 26.0               | 8.0          | 3110       |
| 390 | 327             | 52.0              | 5.3                | 17.7         | 553        |

391 rows x 5 columns

```
[ ] datos_seleccionados.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 391 entries, 0 to 390
Data columns (total 5 columns):
 #   Column              Non-Null Count  Dtype  
---  --
 0   Calorías (kcal)      391 non-null    int64  
 1   Carbohidratos (g)    391 non-null    float64
 2   Lípidos/grasas (g)   391 non-null    float64
 3   Proteína (g)         391 non-null    float64
 4   Sodio (mg)           391 non-null    int64  
dtypes: float64(3), int64(2)
memory usage: 15.4 KB

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

dataset = datos_seleccionados.dropna() # creamos un nuevo dataframe descartando los valores nulos o vacíos de nuestro dataframe datos_seleccionados

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

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

X = dataset[['Carbohidratos (g)', 'Lípidos/grasas (g)', 'Proteína (g)', 'Sodio (mg)']].values # variables independientes
y = dataset['Calorías (kcal)'].values # variable dependiente

[ ] 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% para entrenamiento y 20% para prueba

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

modelo_regresion = LinearRegression() # modelo de regresión

```

## Evidencia 2. Proyecto de Ciencia de Datos

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

LinearRegression()

[ ] 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 coeficientes y sus valores; por cada unidad del coeficiente, su impacto en las calorías será igual a su valor
```

| Coeficientes       |           |
|--------------------|-----------|
| Carbohidratos (g)  | 4.142930  |
| Lípidos/grasas (g) | 9.060501  |
| Proteína (g)       | 3.985310  |
| Sodio (mg)         | -0.011956 |

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

validation = pd.DataFrame({'Actual': y_test, 'Predicción': y_pred, 'Diferencia': y_test-y_pred}) # creamos un dataframe con los valores actuales y los de predicción
muestra_validacion = validation.head(25) # elegimos una muestra con 25 valores
muestra_validacion # desplegamos esos 25 valores
```

|    | Actual | Predicción | Diferencia |
|----|--------|------------|------------|
| 0  | 442    | 442.775806 | -0.775806  |
| 1  | 180    | 196.270902 | -16.270902 |
| 2  | 343    | 339.926061 | 3.073939   |
| 3  | 572    | 576.454573 | -4.454573  |
| 4  | 140    | 137.308400 | 2.691600   |
| 6  | 130    | 127.873612 | 2.126388   |
| 8  | 406    | 501.111034 | -95.111034 |
| 7  | 130    | 127.873612 | 2.126388   |
| 8  | 572    | 576.454573 | -4.454573  |
| 9  | 401    | 372.666362 | 28.333638  |
| 10 | 276    | 269.844275 | 6.155725   |
| 11 | 180    | 196.270902 | -16.270902 |
| 12 | 122    | 114.111317 | 7.888683   |
| 13 | 111    | 104.384859 | 6.615141   |
| 14 | 276    | 269.844275 | 6.155725   |
| 16 | 140    | 137.308400 | 2.691600   |
| 18 | 252    | 250.959528 | 1.040472   |
| 17 | 130    | 123.901455 | 6.098545   |

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

count      79.000000
mean        1.176466
std         42.356121
min        -95.111034
25%        -3.895230
50%         2.691600
75%         7.564434
max        285.238360
Name: Diferencia, dtype: float64

[ ] 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

0.9461940602031965

[ ] 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 y de predicción

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
```



```
import pandas as pd # importa la librería pandas y la asigna a la variable pd

datos_consumo = pd.read_excel('A01253031_Registro-1_Avance.xlsx') # indicamos
el nombre de nuestro archivo a ser leído

datos_consumo.groupby("Momento").count() # con la función groupby agrupamos
los datos de la columna Momento y con count() los contamos para obtener
subtotales

datos_consumo.describe()

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

```
datos_seleccionados # desplegamos el dataframe

datos_seleccionados.info()

datos_seleccionados.isnull().values.any() # buscamos valores nulos y obtenemos
True o False dependiendo si hay o no

dataset = datos_seleccionados.dropna() # creamos un nuevo dataframe
descartando los valores nulos o vacíos de nuestro dataframe datos_seleccionados

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

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

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

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

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% para entrenamiento y 20% para prueba

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

```

modelo_regresion = LinearRegression() # modelo de regresión

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

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 coeficientes y sus valores; por cada unidad del coeficiente, su
impacto en las calorías será igual a su valo

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

validacion = pd.DataFrame({'Actual': y_test, 'Predicción': y_pred, 'Diferencia':
y_test-y_pred}) # creamos un dataframe con los valores actuales y los de predicción

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

muestra_validacion # desplegamos esos 25 valores

validacion["Diferencia"].describe()

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

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 y de predicción

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
```

Lo que hice fue básicamente importar la librería Pandas para poder analizar los datos, ya que la importe utilicé la función groupby, para agrupar los datos de la columna Momento y con la función count contamos la cantidad de momentos. Después obtuvimos una estadística descriptiva para poder seleccionar los datos. Hice una variable datos para asignarle únicamente los datos que iba a analizar, para proceder a limpiar los datos. Utilice varias funciones para buscar valores nulos y se creó un nuevo dataframe con los datos que no sean nulos. Después de ahí le asigne la variable X a los atributos de entrada y Y a los de salida. Dividí mis datos uno en un conjunto de entrenamiento y otro conjunto de prueba, modele los datos y ya nomas visualice los datos para hacer una comparación.



## **Parte 2: Modelación de los datos**

Los datos que se analizan por lo general se procesan utilizando herramientas tecnológicas como lenguajes de programación, ya sea Python, Java, C++ entre otros. Casi siempre se ejecutan varios modelos y luego se deben ajustar dichos parámetros. Por lo general hay casos en los cuales hay modelos que más se adaptan a algunas ocasiones en específico, pero para poder saber con más exactitud se debe examinar los resultados de cada modelo utilizado y se debe evaluar el modelo. Para cada modelo se debe realizar una evaluación basado en los criterios que se tengan, esto podría ser muy útil como una base, luego para cada modelo se puede generar una lista de resultados y apoyarlo con el uso de gráficos para tener un mejor análisis de los resultados. Otro punto importante es que los resultados deben de tener un sentido lógico. De ahí seguiría clasificar los modelos según ciertos criterios, como lo podrían ser los objetivos, precisión del modelo y subjetivos y facilidad de uso o interpretación de los resultados. Con base a los resultados que se tienen es importante realizar una revisión exhaustiva de estos, consultar si es posible con otros analistas de datos, considerar si los resultados son fáciles de desplegar y por último comprobar si los resultados cumplen con los objetivos del problema o lo que se quiera resolver.

### **1. ¿Cuántos intentos o corridas realizaste para obtener los resultados sin errores? Porque**

Gracias a que desde que empecé a registrar los datos y me percate de que tenía que ser más objetivo con mis registros, yo creo que no ocupe más que 1 intento para obtener los resultados sin errores y yo creo que por eso es por lo que no ocupe otro intento.

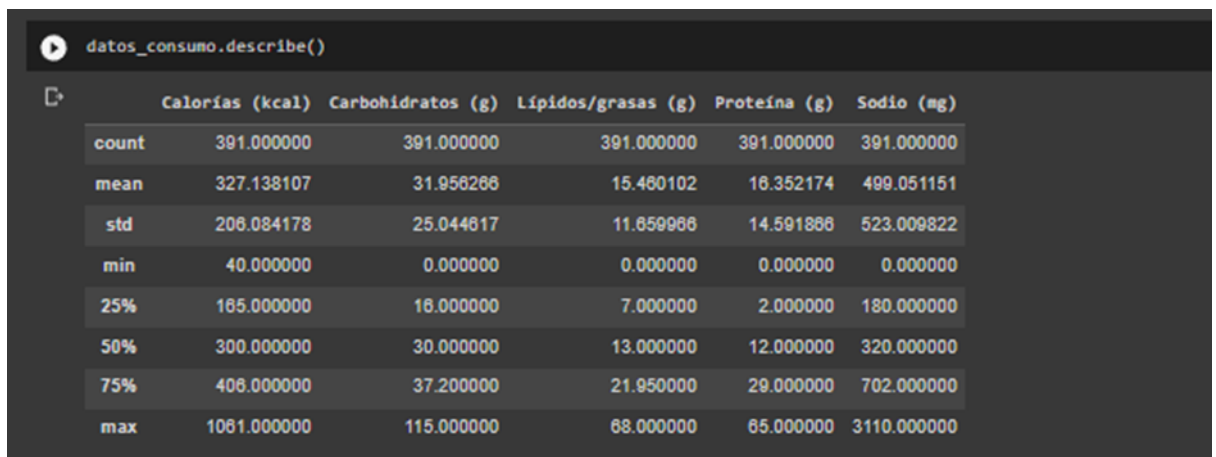
### **2. ¿Cómo los resolviste los problemas que se presentaron?**

Para poder resolver los problemas que se presentaron fue consultarlo con mis compañeros y comparar con los datos que a ellos les arrojaban y en

base a eso, podía tener un mejor panorama del problema que tenía y así intentar replicar el camino que ellos realizaron para que mi análisis sea más exacto, aunque sus datos fueran diferentes.

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

#### 1. Estadística descriptiva

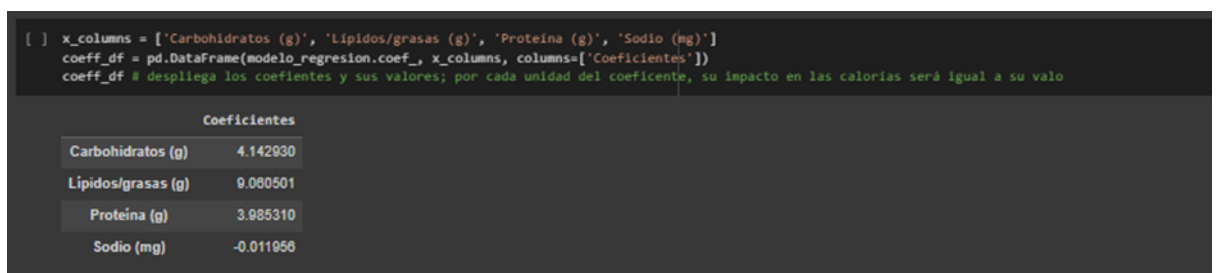


```
datos_consumo.describe()
```

|       | Calorías (kcal) | Carbohidratos (g) | Lípidos/grasas (g) | Proteína (g) | Sodio (mg)  |
|-------|-----------------|-------------------|--------------------|--------------|-------------|
| count | 391.000000      | 391.000000        | 391.000000         | 391.000000   | 391.000000  |
| mean  | 327.138107      | 31.956268         | 15.460102          | 16.352174    | 499.051151  |
| std   | 206.084178      | 25.044617         | 11.859988          | 14.591888    | 523.009822  |
| min   | 40.000000       | 0.000000          | 0.000000           | 0.000000     | 0.000000    |
| 25%   | 185.000000      | 16.000000         | 7.000000           | 2.000000     | 180.000000  |
| 50%   | 300.000000      | 30.000000         | 13.000000          | 12.000000    | 320.000000  |
| 75%   | 406.000000      | 37.200000         | 21.950000          | 29.000000    | 702.000000  |
| max   | 1061.000000     | 115.000000        | 68.000000          | 65.000000    | 3110.000000 |

Lo que se observa aquí es básicamente es una estadística descriptiva con solo los nutrientes en el cual se realizan varios análisis como el mínimo y el máximo entre otros resultados más que me ayudan al análisis que se quiere realizar.

#### 2. Coeficientes de regresión



```
[ ] 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 coeficientes y sus valores; por cada unidad del coeficiente, su impacto en las calorías será igual a su valor
```

| Coeficientes       |           |
|--------------------|-----------|
| Carbohidratos (g)  | 4.142930  |
| Lípidos/grasas (g) | 0.080501  |
| Proteína (g)       | 3.985310  |
| Sodio (mg)         | -0.011958 |

Aquí el algoritmo ya ha aprendido cuáles son los coeficientes de X óptimos para satisfacer el modelo.

### 3. Valores actuales y de predicción

```
[ ] validation = pd.DataFrame({'Actual': y_test, 'Predicción': y_pred, 'Diferencia': y_test-y_pred}) # creamos un dataframe con los valores actuales y los de predicción
mostrar_validation = validation.head(25) # elegimos una muestra con 25 valores
mostrar_validation # desplegamos esos 25 valores
```

|    | Actual | Predicción | Diferencia |
|----|--------|------------|------------|
| 0  | 442    | 442.775806 | -0.775806  |
| 1  | 180    | 196.270902 | -16.270902 |
| 2  | 343    | 339.926061 | 3.073939   |
| 3  | 572    | 576.454573 | -4.454573  |
| 4  | 140    | 137.308400 | 2.691600   |
| 5  | 130    | 127.873612 | 2.126388   |
| 6  | 406    | 501.111034 | -95.111034 |
| 7  | 130    | 127.873612 | 2.126388   |
| 8  | 572    | 576.454573 | -4.454573  |
| 9  | 401    | 372.666362 | 28.333638  |
| 10 | 276    | 269.844275 | 6.155725   |
| 11 | 180    | 196.270902 | -16.270902 |
| 12 | 122    | 114.111317 | 7.888683   |
| 13 | 111    | 104.384809 | 6.615141   |
| 14 | 276    | 269.844275 | 6.155725   |
| 15 | 140    | 137.308400 | 2.691600   |
| 16 | 252    | 250.959528 | 1.040472   |
| 17 | 130    | 123.901455 | 6.098545   |
| 18 | 696    | 699.335887 | -3.335887  |
| 19 | 696    | 699.335887 | -3.335887  |
| 20 | 276    | 269.844275 | 6.155725   |
| 21 | 401    | 372.666362 | 28.333638  |
| 22 | 690    | 404.761640 | 285.238360 |
| 23 | 406    | 501.111034 | -95.111034 |
| 24 | 511    | 448.904655 | 62.095345  |

Se generó una tabla con una comparación de los valores actuales y de predicción en el cual se muestran 25 valores y la diferencia que existe entre los valores.

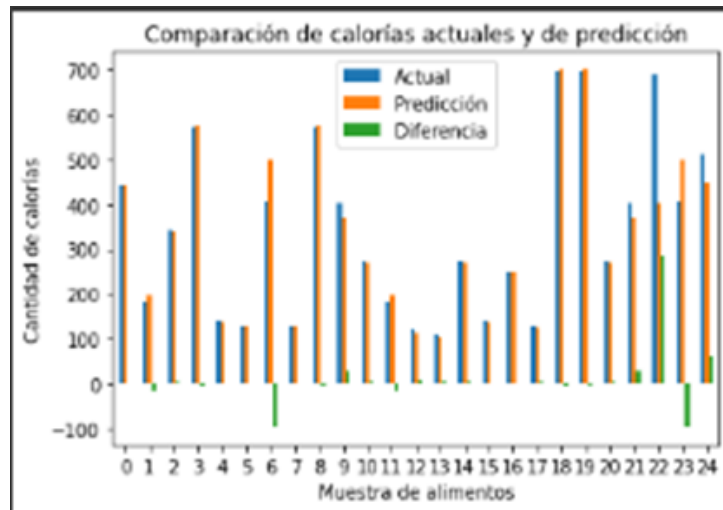
### 4. Coeficiente de determinación r2

```
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
```

0.9461940602031965

Luego aquí tenemos el coeficiente de determinación de r2, que no es muy útil, ya que entre mayor sea el R2, mejor será el ajuste del modelo a los datos. Se espera que sea un valor lo más cercano a 1 por lo cual a mi parecer es muy cercano.

### 5. Gráfica



Lo que obtuve en esta gráfica fue la comparación de calorías actuales y de predicción, en los cuales si se puede observar hay 3 colores en la tabla, una representa las calorías actuales, otro la predicción con naranja y por último el verde con la diferencia que existe entre los datos actuales y a los de predicción.

#### 4. ¿Cuáles son tus conclusiones de la modelación?

En conclusión utilizar herramientas tecnológicas como Python son esenciales para la modelación de datos, ya que estos mismos nos ayudan a tener un mejor análisis de los datos y tener muchos mejores resultados que sean entendibles para poder resolver el problema que se intenta resolver.

Link google colab:

<https://colab.research.google.com/drive/1kNiUYiZwulTU63b9e-OMhh6HloDREPa9?usp=sharing>

## Efecto del consumo calórico

```
[ ] import pandas as pd # importa la librería pandas y la asigna a la variable pd

[ ] datos_consumo = pd.read_excel('A01253031_Registro-1_Avance.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 | 2022-08-17       | Desayuno | Huevo con tortilla       | 303             | 23.0              | 17.0               | 16.0         | 233        | MyFitnessPal |
| 1 | 2022-08-17       | Comida   | Pollo asado              | 422             | 2.0               | 19.0               | 60.0         | 2          | MyFitnessPal |
| 2 | 2022-08-17       | Cena     | Quesadillas con frijoles | 572             | 51.0              | 29.0               | 29.0         | 702        | MyFitnessPal |
| 3 | 2022-08-17       | Snack    | Nilo                     | 252             | 37.0              | 10.0               | 3.0          | 10         | MyFitnessPal |
| 4 | 2022-08-17       | Snack    | 5 Picalfresas            | 100             | 25.0              | 0.0                | 0.0          | 225        | MyFitnessPal |

```
[ ] 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 | 2022-08-17       | 303             |
| 1 | 2022-08-17       | 422             |
| 2 | 2022-08-17       | 572             |
| 3 | 2022-08-17       | 252             |
| 4 | 2022-08-17       | 100             |

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

127911

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

78

[ ] calorías_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:", calorías_promedio)

Tu promedio de calorías consumidas en 78 días es: 1639.8846153846155
```

```
[ ] peso = int(input("Ingresa tu peso en kilogramos: "))
altura = int(input("Ingresa tu altura en centímetros: "))
edad = int(input("Ingresa tu edad en años: "))
genero = input("Ingresa tu género, Mujer/Hombre: ")

Ingresa tu peso en kilogramos: 71
Ingresa tu altura en centímetros: 176
Ingresa tu edad en años: 18
Ingresa tu género, Mujer/Hombre: Hombre
```

```
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: 1800.35
```

```
[ ] diferencia = calorías_promedio - calorías_requeridas

diferencia

-160.46538461538444

[ ] efecto_anual = diferencia * 450/3500 * 365 /1000 # realiza la proporción, se multiplica por 365 (días) y se divide entre 1000 (gramos) para obtener kilogramos
print("Si continúas con el consumo calórico actual, en un año tu cambio de masa corporal sería aproximadamente de:", efecto_anual, "kg")

Si continúas con el consumo calórico actual, en un año tu cambio de masa corporal sería aproximadamente de: -7.538411263736255 kg
```

**import pandas as pd # importa la librería pandas y la asigna a la variable pd**

```

datos_consumo = pd.read_excel('A01253031_Registro-1_Avance.xlsx') #
indicamos el nombre de nuestro archivo a ser leído
datos = datos_consumo[["Fecha (dd/mm/aa)", "Calorías (kcal)"]] #
seleccionamos las dos columnas que necesitaremos
datos.head() # imprimiendo los datos seleccionados
suma_calorias = datos["Calorías (kcal)"].sum()
suma_calorias # despliega el total de calorías
dias = datos["Fecha (dd/mm/aa)"].nunique()
dias # despliega el total de días únicos
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)
peso = int(input("Ingresa tu peso en kilogramos: "))

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

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

genero = input("Ingresa tu género, Mujer/Hombre: ")
if(genero == "Mujer"):
    calorias_requeridas = 655+(9.56*peso)+(1.85*altura)-(4.68*edad) # fórmula
para estimar calorías requeridas en mujer

elif(genero == "Hombre"):
    calorias_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:",
calorias_requeridas)
diferencia = calorias_promedio - calorias_requeridas

diferencia

```

```
efecto_anual = diferencia * 450/3500 * 365 /1000 # realiza la proporción, se
multiplica por 365 (días) y se divide entre 1000 (gramos) para obtener
kilogramos

print("Si continuas con el consumo calórico actual, en un año tu cambio de
masa corporal sería aproximadamente de:",efecto_anual,"kg")
```

Aquí básicamente lo que hice con la función sum, calcule el número total de calorías consumidas, luego con la función nunique conte el total de días diferentes de consumo de calorías. Después calcule el promedio de calorías, por consiguiente agregue varios inputs para que el usuario ingrese sus datos, con este realice a realizar la estimación de calorías requeridas diarias de acuerdo a los datos que se ingresaba. Luego calcule la diferencia entre las calorías consumidas y las requeridas y dependiendo de esto indica si mi consumo es mayor, menor o igual. Con la diferencia que me arroje el programa voy hacer una aproximación en un año, si el resultado es negativo indica que perderé grasa y si es positivo que ganare masa.

Link google colab:

[https://colab.research.google.com/drive/1be-gbvr8Y6dI2X1cU0us8sPQ1\\_3y0pAN?usp=sharing](https://colab.research.google.com/drive/1be-gbvr8Y6dI2X1cU0us8sPQ1_3y0pAN?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.  
Si, porque con la ecuación de Harris - Benedict es ideal para conocer cuál es la cantidad perfecta de calorías, por lo cual de acuerdo a los resultados la estimación se acepta y con lo que me arroja que debería estar en déficit y con esto debería estar bajando 7 kg.
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

## Evidencia 2. Proyecto de Ciencia de Datos

para explicar las diferencias, incluye imagen y explicación de cada resultado en Excel y Python. ¿Cuál te pareció mejor, por qué?

### Excel

#### Estadística descriptiva

La estadística que se muestra en excel es mucho más clara, ya que hay más descripción y se entiende mejor, se pueden ver los valores estadísticos de los nutrientes como media, máximo, mínimo entre otros.

| Calories (kcal)    | Carbohydrates (g) |                    |             | Lipids/grasses (g) | Proteins (g) | Sodium (mg)        |            |
|--------------------|-------------------|--------------------|-------------|--------------------|--------------|--------------------|------------|
| Mean               | 324.154444        | Mean               | 30.82325333 | Mean               | 15.91666667  | Mean               | 481.8      |
| Standard Error     | 38.3479043        | Standard Error     | 4.07857607  | Standard Error     | 2.096090969  | Standard Error     | 17.33      |
| Median             | 284.5             | Median             | 25          | Median             | 12.5         | Median             | 320        |
| Mode               | 171               | Mode               | 25          | Mode               | 7            | Mode               | 10         |
| Standard Deviation | 233.6868258       | Standard Deviation | 13.83654581 | Standard Deviation | 16.13393333  | Standard Deviation | 455.6      |
| Sample Variance    | 54609.53254       | Sample Variance    | 539.220704  | Sample Variance    | 191.45       | Sample Variance    | 209.051143 |
| Kurtosis           | 2.443744564       | Kurtosis           | 2.10574473  | Kurtosis           | 5.146500447  | Kurtosis           | 1.21       |
| Skewness           | 1.634127435       | Skewness           | 1.93531076  | Skewness           | 1.957898779  | Skewness           | 0.82447478 |
| Range              | 361               | Range              | 115         | Range              | 65           | Range              | 1710       |
| Minimum            | 100               | Minimum            | 0           | Minimum            | 0            | Minimum            | 0          |
| Maximum            | 1061              | Maximum            | 115         | Maximum            | 65           | Maximum            | 65         |
| Sum                | 12021             | Sum                | 1110        | Sum                | 373          | Sum                | 6662       |
| Count              | 36                | Count              | 36          | Count              | 36           | Count              | 36         |

#### Coefficiente de regresión

Estos valores son muy parecidos a los que me mostró en google colab gracias a python. Aquí se muestra los coeficientes de cada valor nutricional.

|                    | Coefficients |
|--------------------|--------------|
| Intercept          | -5.82851446  |
| Carbohidratos (g)  | 4.125116592  |
| Lípidos/grasas (g) | 8.699944828  |
| Proteína (g)       | 4.019337413  |

#### Valores actuales y de predicción

Aquí se muestra la tabla con los valores de predicción de calorías y los residuos que es la diferencia.



## Evidencia 2. Proyecto de Ciencia de Datos

| <i>Predicted Calories (kcal)</i> | <i>Residuals</i> |
|----------------------------------|------------------|
| 301.2576278                      | 1.74237216       |
| 408.8809152                      | 13.11908477      |
| 573.4116167                      | -1.411616719     |
| 245.85826                        | 6.141740037      |
| 97.29940034                      | 2.700599662      |
| 505.9814321                      | -99.98143206     |
| 245.85826                        | 6.141740037      |
| 116.0741479                      | 5.925852073      |
| 194.6637935                      | -14.66379346     |
| 318.9207933                      | -14.92079325     |
| 377.4415477                      | 23.55845229      |
| 267.3829948                      | 8.617005186      |
| 104.4667193                      | 6.533280737      |
| 97.29940034                      | 2.700599662      |
| 700.2073168                      | -4.20731681      |
| 272.3552168                      | 9.644783231      |
| 1039.553337                      | 12.44666294      |
| 104.4667193                      | 6.533280737      |
| 125.9636581                      | 4.036341877      |
| 337.0611372                      | 5.938862839      |
| 485.6205973                      | 18.37940268      |
| 124.5368114                      | 5.463188601      |
| 137.8115845                      | 2.188415537      |
| 451.4259532                      | 2.574046769      |
| 1045.030262                      | 15.96973789      |
| 472.7928624                      | -5.792862374     |
| 104.4667193                      | 6.533280737      |
| 434.5307089                      | 7.469291068      |
| 261.6463395                      | 25.35366054      |
| 451.4259532                      | 2.574046769      |
| 452.5132849                      | 1.486715128      |
| 224.0971734                      | -59.09717342     |
| 104.4667193                      | 6.533280737      |
| 245.85826                        | 6.141740037      |
| 318.9207933                      | -14.92079325     |
| 281.4516834                      | -1.451683387     |

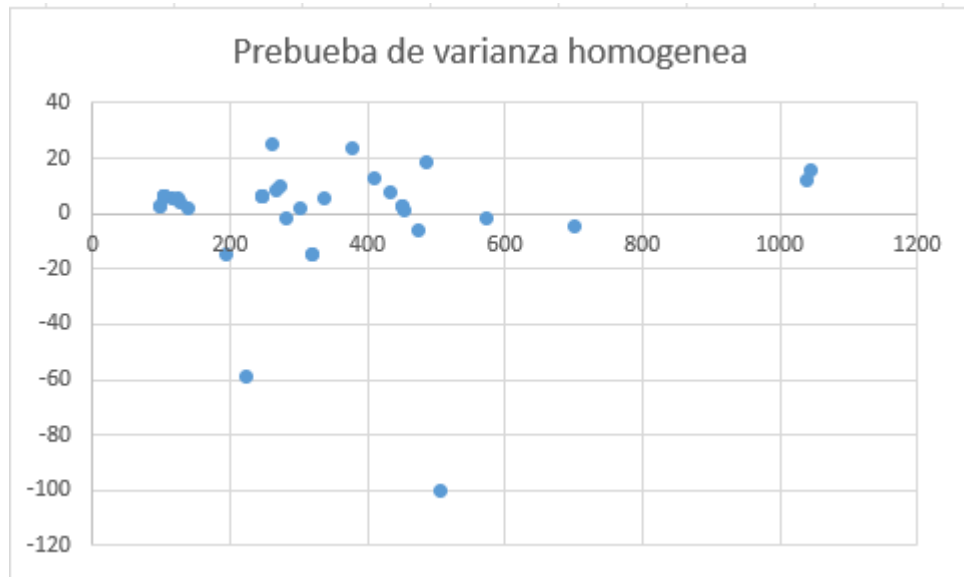
Coeficiente de determinación  $r^2$

Aquí se muestra que el coeficiente de determinación  $r^2$  es 99%, esto porque en excel se tienen algunos datos extras.

| <i>Regression Statistics</i> |             |
|------------------------------|-------------|
| Multiple R                   | 0.998514083 |
| R Square                     | 0.997030374 |
| Adjusted R Square            | 0.965594773 |
| Standard Error               | 23.2803984  |
| Observations                 | 35          |

Gráfica (de la prueba 2)

Aquí se muestra una gráfica de prueba de varianza homogénea que demuestra que tanto varían las proyecciones con los valores reales. Se puede observar que están alejados del eje x.



## Python

### Estadística descriptiva

Aquí la gráfica en google colab es menos específica, ya que tiene menos descripción. Nomas se muestra los datos nutricionales con sus respectivas estadísticas.

```
datos_consumo.describe()
```

|       | Calorias (kcal) | Carbohidratos (g) | Lípidos/grasas (g) | Proteína (g) | Sodio (mg)  |
|-------|-----------------|-------------------|--------------------|--------------|-------------|
| count | 391.000000      | 391.000000        | 391.000000         | 391.000000   | 391.000000  |
| mean  | 327.138107      | 31.556286         | 15.460102          | 10.352174    | 499.051151  |
| std   | 206.084178      | 25.044817         | 11.659966          | 14.591866    | 523.006822  |
| min   | 40.000000       | 0.000000          | 0.000000           | 0.000000     | 0.000000    |
| 25%   | 165.000000      | 16.000000         | 7.000000           | 2.000000     | 180.000000  |
| 50%   | 300.000000      | 30.000000         | 13.000000          | 12.000000    | 320.000000  |
| 75%   | 408.000000      | 37.200000         | 21.950000          | 29.000000    | 702.000000  |
| max   | 1061.000000     | 115.000000        | 68.000000          | 65.000000    | 3110.000000 |

### Coefficiente de regresión

Por lo que aquí se tomaron más datos, existe una mínima diferencia, pero no muy alejada a los números que arroja en excel.

```
[ ] 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 coeficientes y sus valores; por cada unidad del coeficiente, su impacto en las calorías será igual a su valo
```

| Coeficientes       |           |
|--------------------|-----------|
| Carbohidratos (g)  | 4.142930  |
| Lípidos/grasas (g) | 9.080501  |
| Proteína (g)       | 3.985310  |
| Sodio (mg)         | -0.011956 |

### Valores actuales y de predicción

Para poder sacar la diferencia, aquí compararon los valores actuales con las predicciones para poder sacar la diferencia.

```
[ ] validation = pd.DataFrame({'Actual': y_test, 'Predicción': y_pred, 'Diferencia': y_test-y_pred}) # creamos un dataframe con los valores actuales y los de predicción
muestra_validacion = validation.head(25) # elegimos una muestra con 25 valores
muestra_validacion # desplegamos esos 25 valores
```

|    | Actual | Predicción | Diferencia |
|----|--------|------------|------------|
| 0  | 442    | 442.775806 | -0.775806  |
| 1  | 180    | 196.270902 | -16.270902 |
| 2  | 343    | 339.926061 | 3.073939   |
| 3  | 572    | 576.454573 | -4.454573  |
| 4  | 140    | 137.308400 | 2.691600   |
| 5  | 130    | 127.873612 | 2.126388   |
| 6  | 406    | 501.111034 | -95.111034 |
| 7  | 130    | 127.873612 | 2.126388   |
| 8  | 572    | 576.454573 | -4.454573  |
| 9  | 401    | 372.666362 | 28.333638  |
| 10 | 276    | 269.844275 | 6.155725   |
| 11 | 180    | 196.270902 | -16.270902 |
| 12 | 122    | 114.111317 | 7.888683   |
| 13 | 111    | 104.364809 | 6.635191   |
| 14 | 276    | 269.844275 | 6.155725   |
| 15 | 140    | 137.308400 | 2.691600   |
| 16 | 252    | 250.999528 | 1.040472   |
| 17 | 130    | 123.901455 | 6.098545   |
| 18 | 696    | 699.335887 | -3.335887  |
| 19 | 696    | 699.335887 | -3.335887  |
| 20 | 276    | 269.844275 | 6.155725   |
| 21 | 401    | 372.666362 | 28.333638  |
| 22 | 690    | 404.761640 | 285.238360 |
| 23 | 406    | 501.111034 | -95.111034 |
| 24 | 511    | 448.904655 | 62.095345  |

### Coeficiente de determinación r2

Aquí el porcentaje cambia 94%, por lo cual es menos preciso esto puede ser por varios factores, pero en lo personal como lo muestra en google colab es más entendible porque solo muestra el dato que queremos conocer qué es el coeficiente de determinación r2.

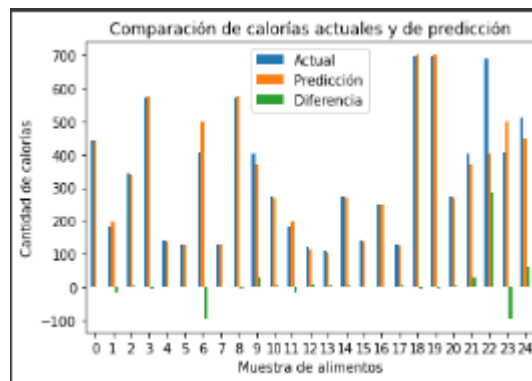
```
from sklearn.metrics import r2_score # importamos la métrica R cuadrada (coeficiente de determinación)

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

0.9461940602031965
```

### Gráfica (de barras)

La gráfica que se muestra es mucho más fácil de entender, ya que esta muestra más descriptores y con su color específico. Aquí se representan las diferencias entre las calorías reales y las obtenidas por el modelo, se incluyen todos los valores nutricionales.



### 3. ¿Por qué es importante la Ciencia de Datos y la ética para el uso adecuado de los datos e información?

El uso adecuado de datos e información es muy importante cuando se trata del uso de grandes cantidades de información en el cual se utiliza la ciencia de datos, ya que la ciencias de datos puede ser muy útil pero si cae en mano equivocadas puede ser utilizado de mala manera y afectar a muchas gente, por lo cual la ética dentro de esta área es muy importante para que no sucedan situaciones en las cuales gente aproveche esta herramienta para su beneficio. (Calle, 2018)

Como lo fue el caso de los usuarios de Facebook, ya que con la ayuda de la minería de datos y el análisis de datos pudieron explotar la información personal de los

usuarios de esta plataforma y así poder utilizarlo a su favor en las elecciones. Esto además atenta contra las políticas de uso de la red social y luego estos datos fueron utilizados para crear anuncios políticos durante las elecciones presidenciales. Yo creo que ambas partes estuvieron muy mal, ya que Meta Platforms, la empresa matriz de Facebook e Instagram, dejó que estos fueran revelados. ( Los Angeles Times, 2022)

Referencias:

-Amazon. (2022). *¿Qué es la ciencia de datos? - Guía sobre la ciencia de datos para principiantes* - AWS. Amazon Web Services, Inc.

<https://aws.amazon.com/es/what-is/data-science/>

-Calle, C. (2018, 19 mayo). *¿Cómo aplicamos la ética al Big Data y la Inteligencia Artificial?* KPMG Tendencias.

<https://www.tendencias.kpmg.es/2018/04/etica-big-data/>

-Los Angeles Times (2022, 27 agosto). *Meta llega a acuerdo en demanda por caso Cambridge Analytica* - Los Angeles Times.. Recuperado 16 de octubre de 2022, de <https://www.latimes.com/espanol/eeuu/articulo/2022-08-27/meta-llega-a-acuerdo-en-demanda-por-caso-cambridge-analytica>