



Tecnológico
de Monterrey

**Instituto Tecnológico y de Estudios Superiores de
Monterrey**

Myrna Dinorah Ortiz Ramirez

Matrícula: A01383998

Evidencia 2: Proyecto de Ciencia de Datos

Fecha de elaboración: 24 de noviembre del 2020

Introducción

La ciencia de datos es una combinación de diversas áreas, áreas que incluyen estadística, métodos científicos, análisis de datos, entre otros, la ciencia de datos ayuda a resolver las necesidades en el manejo actual y futuro de la información pues es una disciplina que estudia el dónde viene determinada información, la ciencia de datos puede ayudar a una persona en el ámbito laboral y estudiantil pues es bien sabido que cada vez somos más parte de la era de la tecnología. En el ámbito laboral nos puede ayudar con la recopilación de datos de una cierta empresa, con las estadísticas mostradas anteriormente en la susodicha, se pueden mejorar significativamente las ventas creando modelos en base a hipótesis. La ciencia de datos ayuda a tomar decisiones más precisas, informadas y con mayor nivel de detalle de las que antes eran capaces de realizar.

La intención de este proyecto es responder a nuestra hipótesis **¿Si consumo cierta cantidad calórica, puedo tener cambios en mi masa corporal (peso) en un determinado tiempo?**, para esto tuvimos que hacer una recolección de datos en 11 semanas, en las cuales tenías que recopilar información de tu consumo alimenticio día a día, después teníamos que investigar para ver qué era lo que íbamos a hacer para que nuestro proyecto se llevara correctamente, para esto creamos un modelo en Excel y en Google Colab para realizar nuestro modelo en base a la programación, por lo tanto tuvimos que documentar los datos semana a semana. Todo esto se hizo para responder a nuestra hipótesis inicial, pero ¿en qué se basó esta hipótesis?, sabemos que México ocupa el segundo lugar en obesidad a nivel mundial, cuando comparamos a México con Japón notamos una gran diferencia, esto es debido a que en Japón hay tanto restaurantes de comida rápida como restaurantes de comida casera, esto ayuda a concientizar sobre el consumo alimenticio, mientras que en México hay mucho más restaurantes de comida rápida, también en tiendas pequeñas como Oxxo, que nos ofrece una variedad mayor de comida chatarra, y si tienes suerte encontraras algo saludable. La intención del proyecto es ayudar a concientizar a los mexicanos en los alimentos que consumimos día a día.

Fase 1. Entendimiento del negocio

1.- ¿Quién es el cliente?

Yo soy el cliente del proyecto.

2.- ¿Qué problemas estás tratando de resolver?

El problema por resolver es determinar el cambio de masa corporal con base a mi alimentación de 11 semanas

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

La recolección de datos nos proporciona a nosotros una gran cantidad de información sobre nuestro consumo, entre más alimentos se registren más exactas son nuestras conclusiones a partir de esto tenemos como solución la regresión en los que relacionamos variables como proteínas, grasas y carbohidratos con las calorías y se creó un modelo en el que predecía las calorías de un alimento en base a su proteína, grasa y carbohidrato, creamos histogramas que son una representación gráfica, y lo hicimos con cada uno de los datos, esto nos ayuda a visualizar mejor cada función(proteínas, grasas, etc.). Con el programa Excel también obtuvimos líneas de tendencias que nos ayudaban a predecir, y en base a estas en un lapso de 2 semanas de recolección de datos, podemos analizar y crear un modelo más exacto.

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

La solución será a través de un análisis de regresión lineal multivariable, y para lograrlo se necesita aprender estadística, programación en Python y los datos de consumo, así como, aprender la comprobación, interpretar gráficas, aceptar o rechazar hipótesis y sacar conclusiones.

5.- ¿Qué deberás hacer para desarrollar tu solución?

Para algo más general se debe pensar en SMART qué nos adentra más en la realización del proyecto y como consecuencia ser mas organizados en este. En

SMART se nos presenta sencillas preguntas que tenemos que responder antes de adentrarnos a la realización del proyecto.

Debemos estar conscientes de la información que se nos presente, y si presenta alguna anomalía presentarla en nuestras conclusiones y ver como está afecta a nuestros resultados.

Después desarrollamos el proyecto en base a la metodología de las 4 fases.

Entendimiento del negocio, en el que se responden varias preguntas acerca del negocio, fases, su ciencia de datos, etc. En este modelo tenemos que responder las preguntas respecto al informe calórico que se llevó a cabo, como el qué puede conllevar alimentarte sano en un futuro.

Entendimiento de los datos, en esta parte se deben tener bien cimentados los conceptos de estadística para poder aplicarlos en nuestro modelo.

Preparación de los datos, se tiene que ver los errores de cada dato si es que los hay, cuáles datos se irán a utilizar, cuáles son relevantes y cuáles no.

Modelación de los datos, se llevará a cabo el proceso ya planeado con nuestros datos obtenidos del programa Excel.

Para algo más específico se piensa en el modelo CRISP-DM en el que consiste en

Fase I. Business Understanding. Definición de necesidades del cliente (comprensión del negocio)

Esta fase inicial se enfoca en la comprensión de los objetivos de proyecto.

Después se convierte este conocimiento de los datos en la definición de un problema en este caso la obesidad en México así que hacemos un plan preliminar diseñado para alcanzar los objetivos.

Fase II. Data Understanding. Estudio y comprensión de los datos

La fase de entendimiento de datos comienza con la colección de datos inicial y continúa con las actividades que permiten familiarizarse con los datos en este caso fue la recolección de datos en 11 semanas.

Fase III. Data Preparation. Análisis de los datos y selección de características

La fase de preparación de datos cubre todas las actividades necesarias para construir el conjunto final de datos (los datos que se utilizarán en las herramientas de modelado) a partir de los datos en bruto iniciales. Las tareas incluyen la selección de tablas, registros y atributos, así como la transformación y la limpieza de datos para las herramientas que modelan. En nuestro caso usamos Excel para llevar a cabo la selección de tablas, registros y atributos.

Fase IV. Modeling. Modelado

En esta fase, se seleccionan y aplican las técnicas de modelado que sean pertinentes al problema (cuantas más mejor), y se calibran sus parámetros a valores óptimos.

Algunas técnicas tienen requerimientos específicos sobre la forma de los datos.

Fase V. Evaluation. Evaluación (obtención de resultados)

En esta etapa en el proyecto, se han construido uno o varios modelos que parecen alcanzar calidad suficiente desde la una perspectiva de análisis de datos.

Antes de proceder al despliegue final del modelo, es importante evaluarlo a fondo y revisar los pasos ejecutados para crearlo, comparar el modelo obtenido con los objetivos de negocio. Un objetivo clave es determinar si hay alguna cuestión importante de negocio que no haya sido considerada suficientemente. Al final de

esta fase, se debería obtener una decisión sobre la aplicación de los resultados del proceso de análisis de datos.

Fase VI. Deployment. Despliegue (puesta en producción)

Generalmente, la creación del modelo no es el final del proyecto. Incluso si el objetivo del modelo es de aumentar el conocimiento de los datos, el conocimiento obtenido tendrá que organizarse y presentarse para que el cliente pueda usarlo. Dependiendo de los requisitos, la fase de desarrollo puede ser tan simple como la generación de un informe o tan compleja como la realización periódica y quizás automatizada de un proceso de análisis de datos en la organización

Fase 2. Entendimiento de los datos

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

Los datos existentes son los que obtuve a partir de Fatsecret, lo que se consumió que serían las calorías, los datos adquiridos, son los máximos, los mínimos, los porcentajes, etc y los datos adicionales son el tipo de datos dtypes.

Nuestros datos

En este caso estamos usando datos existentes en base a la recolección de datos que hicimos en estas semanas, obteniendo así la fecha, el momento, el nombre del alimento, las calorías, los carbohidratos, los lípidos/grasas, la proteína y el sodio.

2.- ¿Qué tipos de datos se analizarán?

Existen dos tipos de datos que son los string que son las letras o palabras y los float que son los valores numéricos, como ejemplos de los datos que tenemos en nuestro modelo serían la fecha, el momento, el nombre del alimento, las calorías, los carbohidratos, los lípidos/grasas, la proteína y el sodio.

En nuestro caso solo analizaremos los datos numéricos enteros y de punto flotante según lo que reporta Python y continuos de proporción ya que el cero significa ausencia de la característica

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

Las columnas prometedoras son las de calorías, carbohidratos, lípidos y proteínas, pero el atributo más prometedor es el de calorías

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

Los atributos irrelevantes podrían ser los que no se incluyen en el análisis como fecha, nombre del alimento, momento de consumo y fuente.

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

No, estamos analizando 5 filas de nuestro programa que son las primeras 5 o las ultimas 5 filas, esto no nos aporta mucha información, se debería analizar todo el documento sin variables repetidas. Estadísticamente podemos decir que más de 100 datos son suficientes

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

No, los atributos son para crear un modelo eficiente con las variables correctas y necesarias, no hay demasiados atributos sobre todo considerando que para el modelo se utilizarán solo 5 de los 9 que tiene nuestra base de datos

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?

Se obtuvieron de nuestro modelo anterior, si se están fusionando, se esta fusionando los datos de nuestro Excel que son numéricos y alfabéticos a nuestro programa en Python que los convierte en string y en float, podría haber problemas

al compilar, si el archivo tiene demasiados datos por todas partes, si esta mal escrito o un error al escribir e incluso de indentación.

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

Sí, podemos rellenar los datos faltantes, en este caso sería la fecha y si falta algún valor numérico checar el documento en Excel e inspeccionar el alimento que consumiste en ese día, para mi es mejor modificar el documento que el programa ya que el programa no te puede correr. Los datos faltantes se pueden buscar en otra fuente, poner cero o eliminar el registro para que no afecte a nuestro modelo

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

Tenemos nuestros datos en Excel, la calidad se determina de acuerdo a la revisión y el resultado de su análisis.

Todos los datos de nuestro documento están accesibles y disponibles, y la calidad es la misma para todos, en este caso serian 8 columnas y 305 filas lo que da un total de 2440 datos analizándose en un programa en el que tomaramos en cuenta todo nuestro documento de Excel, en este caso solo estamos analizando 5 filas y 8 columnas que seria un total de 40 datos y todos tienen la misma calidad.

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

La hipótesis del proyecto es : **¿Si consumo cierta cantidad calórica, puedo tener cambios en mi masa corporal (peso) en un determinado tiempo?** Y la relación seria que con nuestros datos queremos predecir la cantidad calórica mediante un programa y crear un modelo en el que si consumes cierta cantidad calórica durante una semana como esto influye en la masa, vamos a resolver nuestra hipótesis con los datos obtenidos anteriormente y así crear un modelo que nos ayude a anticipar y así poder solucionar en caso de que estemos en un IMC de riesgo.

Fase 3. Preparación de los datos

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

Los primeros 5 datos del archivo de Excel, para que sean nuestros datos de entrenamiento, también si queremos usar otros comandos se importan todos los datos del archivo de Excel, solo que en nuestro caso solo usamos los primeros cinco.

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

Hay que reemplazar los valores en blanco ya que si hacemos nuestro programa en Python y validamos que nuestros valores son iguales a cero no podríamos proseguir con nuestra preparación de datos.

En conclusión, en mi programa no tengo que rellenar datos en mi Excel, no hay necesidad de eliminar o reemplazar datos en blanco, lo verificamos con un comando en Python en la fase 2, aunque también en este caso se pueden eliminar registros que tengan datos en blanco para que no afecten los resultados del modelo, si es que hubiese.

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

Sí se puede, haciendo un archivo de Excel nuevo con los datos existentes añadiéndolos en ese archivo, así podríamos importarlo en Python y usarlos, añadiendo datos te beneficiaría ya que el modelo tendría más exactitud.

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

Yo solo use datos de FatSecret, no se usó otras fuentes como etiquetas, u otras fuentes de internet, como consecuencia para este modelo tenemos que los datos utilizados no se fusionaron

Pero si se puede integrar datos de distintas fuentes, aunque no es necesario, ya que no afecta en el modelo matemático, un punto importante es que se tiene que corroborar que la fuente sea confiable y segura para que como consecuencia tengamos datos confiables y sobre todo datos reales.

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

No los vamos a ordenar, queremos que estén desordenados para no crear desproporción o sesgo en el modelo. En este modelo no se requiere un formato u orden particular de los datos.

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

Si, lo hacemos para verificar que tan bien esta nuestro programa y sucesivamente lo utilizamos para comparar los datos reales con los datos de entrenamiento, el conjunto de entrenamiento es para entrenar el algoritmo y el conjunto de prueba es para probar si se entrenó correctamente.

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

Yo tuve que agregar datos porque eran muy pocos los que había hecho, no tuve que cambiar ningún registro en el Excel, solo asigne las variables x y y al final del programa, tampoco modifique ningún dato originalmente. No es necesario hacer cambios en los atributos (columnas) de los datos ya que se tienen los necesarios para realizar el modelo de predicción.

Fase 4. Modelación de los datos**Código**

```

import pandas as pd
datos_consumo = pd.read_excel('datos.xlsx')

datos_consumo.head()
datos_consumo.groupby("Momento").count()
datos_consumo.describe()
datos_seleccionados = datos_consumo.iloc[:,3:8]
datos_seleccionados

datos_seleccionados.info()
datos_seleccionados.isnull().values.any()

dataset = datos_seleccionados.dropna()
dataset.isnull().sum()
dataset.columns
X = dataset[['Carbohidratos (g)', 'Lípidos/grasas (g)', 'Proteína (g)', 'Sodio (mg)']].va
lues
y = dataset['Calorías (kcal)'].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=
0)
from sklearn.linear_model import LinearRegression
modelo_regresion = LinearRegression()
modelo_regresion.fit(X_train, y_train)

```

```

x_columns = ['Carbohidratos (g)', 'Lípidos/grasas (g)', 'Proteína (g)', 'Sodio (mg)']
coeff_df = pd.DataFrame(modelo_regresion.coef_, x_columns, columns=['Coeficientes'])
coeff_df

y_pred = modelo_regresion.predict(X_test)

validacion = pd.DataFrame({'Actual': y_test, 'Predicción': y_pred, 'Diferencia': y_test - y_pred})
muestra_validacion = validacion.head(25)
muestra_validacion
validacion["Diferencia"].describe()
from sklearn import metrics
import numpy as np

print("Raíz de la desviación media al cuadrado:", np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
import matplotlib.pyplot as plt

muestra_validacion.plot.bar(rot=0)

plt.title("Comparación de calorías actuales y de predicción")

plt.xlabel("Muestra de alimentos")
plt.ylabel("Cantidad de calorías")

plt.show()

```

Evidencia 2. Proyecto de Ciencia de Datos

```
[1] import pandas as pd
datos_consumo = pd.read_excel('datos.xlsx')

datos_consumo.head()
```

	Fecha (dd/mm/aa)	Momento	Nombre alimento	Calorías (kcal)	Carbohidratos (g)	Lípidos/grasas (g)	Proteína (g)	Sodio (mg)
0	2020-08-17	Desayuno	Kellogg's Cereal Extra Arándanos	113	25.00	0.50	2.00	110.0
1	2020-08-17	Desayuno	Lala Leche Deslactosada Light	91	12.00	1.30	7.80	116.0
2	2020-08-17	Comida	Arroz con pollo	299	38.01	10.88	11.32	544.0
3	2020-08-17	Cena	Tostadas con Frijoles, Queso, Carne, Lechuga, ...	151	12.54	7.85	8.12	274.0
4	2020-08-18	Desayuno	Kellogg's Cereal Extra Arándanos	113	25.00	0.50	2.00	110.0

```
[2] datos_consumo.groupby("Momento").count()
```

	Fecha (dd/mm/aa)	Nombre alimento	Calorías (kcal)	Carbohidratos (g)	Lípidos/grasas (g)	Proteína (g)	Sodio (mg)
Momento							
Cena	74	74	74	74	74	74	74
Cena	9	9	9	9	9	9	9
Comida	90	90	90	90	90	90	90
Comida	3	3	3	3	3	3	3
Desayuno	126	126	126	126	126	126	126
Snacks	4	4	4	4	4	4	4

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

2	2020-08-17	Comida	Arroz con pollo	299	38.01	10.88	11.32	544.0
3	2020-08-17	Cena	Tostadas con Frijoles, Queso, Carne, Lechuga, ...	151	12.54	7.85	8.12	274.0
4	2020-08-18	Desayuno	Kellogg's Cereal Extra Arándanos	113	25.00	0.50	2.00	110.0

```
[2] datos_consumo.groupby("Momento").count()
```

	Fecha (dd/mm/aa)	Nombre alimento	Calorías (kcal)	Carbohidratos (g)	Lípidos/grasas (g)	Proteína (g)	Sodio (mg)
Momento							
Cena	74	74	74	74	74	74	74
Cena	9	9	9	9	9	9	9
Comida	90	90	90	90	90	90	90
Comida	3	3	3	3	3	3	3
Desayuno	126	126	126	126	126	126	126
Snacks	4	4	4	4	4	4	4

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

	Calorías (kcal)	Carbohidratos (g)	Lípidos/grasas (g)	Proteína (g)	Sodio (mg)
count	306.000000	306.000000	306.000000	306.000000	306.000000
mean	139.160131	16.212222	5.328301	6.628497	185.178039
std	109.790279	12.890608	6.609196	7.406837	269.495056
min	2.000000	0.000000	0.000000	0.000000	0.000000
25%	78.000000	6.000000	0.500000	1.740000	5.000000
50%	114.000000	12.050000	3.400000	4.220000	114.000000

Evidencia 2. Proyecto de Ciencia de Datos

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

	Calorías (kcal)	Carbohidratos (g)	Lípidos/grasas (g)	Proteína (g)	Sodio (mg)
count	306.000000	306.000000	306.000000	306.000000	306.000000
mean	139.160131	16.212222	5.328301	6.628497	185.178039
std	109.790279	12.890608	6.609196	7.406837	269.495056
min	2.000000	0.000000	0.000000	0.000000	0.000000
25%	78.000000	6.000000	0.500000	1.740000	5.000000
50%	114.000000	12.050000	3.400000	4.220000	114.000000
75%	171.000000	24.000000	8.010000	8.810000	211.000000
max	608.000000	75.700000	36.760000	33.000000	1555.000000

```
[6] datos_seleccionados = datos_consumo.iloc[:,3:8]
datos_seleccionados
```

	Calorías (kcal)	Carbohidratos (g)	Lípidos/grasas (g)	Proteína (g)	Sodio (mg)
0	113	25.00	0.50	2.00	110.0
1	91	12.00	1.30	7.80	116.0
2	299	38.01	10.88	11.32	544.0
3	151	12.54	7.85	8.12	274.0
4	113	25.00	0.50	2.00	110.0
...
301	16	4.20	0.00	0.00	0.0
302	2	0.09	0.05	0.28	5.0
303	167	29.70	4.20	2.60	162.0

```
[6] datos_seleccionados = datos_consumo.iloc[:,3:8]
datos_seleccionados
```

	Calorías (kcal)	Carbohidratos (g)	Lípidos/grasas (g)	Proteína (g)	Sodio (mg)
0	113	25.00	0.50	2.00	110.0
1	91	12.00	1.30	7.80	116.0
2	299	38.01	10.88	11.32	544.0
3	151	12.54	7.85	8.12	274.0
4	113	25.00	0.50	2.00	110.0
...
301	16	4.20	0.00	0.00	0.0
302	2	0.09	0.05	0.28	5.0
303	167	29.70	4.20	2.60	162.0
304	57	2.12	3.69	3.47	134.0
305	91	12.10	2.80	4.40	75.0

306 rows × 5 columns

```
[7] datos_seleccionados.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 306 entries, 0 to 305
Data columns (total 5 columns):
#   Column              Non-Null Count  Dtype
---  --
0   Calorías (kcal)      306 non-null    int64
1   Carbohidratos (g)    306 non-null    float64
2   Lípidos/grasas (g)  306 non-null    float64
3   Proteína (g)         306 non-null    float64
4   Sodio (mg)           306 non-null    float64
```

Evidencia 2. Proyecto de Ciencia de Datos

```
[7] datos_seleccionados.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 306 entries, 0 to 305
Data columns (total 5 columns):
 #   Column              Non-Null Count  Dtype  
---  --
 0   Calorías (kcal)      306 non-null    int64  
 1   Carbohidratos (g)    306 non-null    float64
 2   Lípidos/grasas (g)   306 non-null    float64
 3   Proteína (g)         306 non-null    float64
 4   Sodio (mg)           306 non-null    float64
dtypes: float64(4), int64(1)
memory usage: 12.1 KB
```

```
[8] datos_seleccionados.isnull().values.any()

False
```

```
[11] dataset = datos_seleccionados.dropna()
dataset.isnull().sum()

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

```
[12] dataset.columns
```

+ Código + Texto

✓ RAM 100%
Disco 100% Editando ^

```
[11] dataset = datos_seleccionados.dropna()
dataset.isnull().sum()
```

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

```
[12] dataset.columns
```

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

```
[28] X = dataset[['Carbohidratos (g)', 'Lípidos/grasas (g)', 'Proteína (g)', 'Sodio (mg)']].values
y = dataset['Calorías (kcal)'].values
```

```
[15] from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

```
[18] from sklearn.linear_model import LinearRegression
modelo_regresion = LinearRegression()
modelo_regresion.fit(X_train, y_train)
```

```
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
```

Evidencia 2. Proyecto de Ciencia de Datos

```
[28] X = dataset[['Carbohidratos (g)', 'Lípidos/grasas (g)', 'Proteína (g)', 'Sodio (mg)']].values
      y = dataset['Calorías (kcal)'].values
```

```
[15] from sklearn.model_selection import train_test_split
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

```
[18] from sklearn.linear_model import LinearRegression
      modelo_regresion = LinearRegression()
      modelo_regresion.fit(X_train, y_train)
```

```
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
```

Coeficientes	
Carbohidratos (g)	3.958105
Lípidos/grasas (g)	8.789380
Proteína (g)	4.230374
Sodio (mg)	0.002403

```
[21] y_pred = modelo_regresion.predict(X_test)
```

```
[22] validacion = pd.DataFrame({'Actual': y_test, 'Predicción': y_pred, 'Diferencia': y_test-y_pred})
      muestra_validacion = validacion.head(25)
```

+ Código + Texto

✓ RAM
Disco

✎ Editando

```
[19] Carbohidratos (g)    3.958105
      Lípidos/grasas (g)  8.789380
      Proteína (g)      4.230374
      Sodio (mg)        0.002403
```

```
[21] y_pred = modelo_regresion.predict(X_test)
```

```
[22] validacion = pd.DataFrame({'Actual': y_test, 'Predicción': y_pred, 'Diferencia': y_test-y_pred})
      muestra_validacion = validacion.head(25)
      muestra_validacion
```

	Actual	Predicción	Diferencia
0	16	15.940430	0.059570
1	91	91.515541	-0.515541
2	167	165.175815	1.824185
3	500	411.705050	88.294950
4	135	139.219849	-4.219849
5	383	385.579276	-2.579276
6	435	433.116929	1.883071
7	167	165.175815	1.824185
8	100	92.920761	7.079239
9	126	126.451605	-0.451605
10	91	91.515541	-0.515541
11	81	79.020570	1.979430
12	342	334.685215	7.314785

Evidencia 2. Proyecto de Ciencia de Datos

```
+ Código + Texto RAM Disco Editando ^
```

```
[22] 8 100 92.920761 7.079239
      9 126 126.451605 -0.451605
     10 91 91.515541 -0.515541
     11 81 79.020570 1.979430
     12 342 334.685215 7.314785
     13 122 123.201563 -1.201563
     14 160 167.169714 -7.169714
     15 91 91.515541 -0.515541
     16 114 112.648963 1.351037
     17 121 119.491541 1.508459
     18 171 169.181588 1.818412
     19 16 15.940430 0.059570
     20 122 123.201563 -1.201563
     21 110 106.818565 3.181435
     22 128 125.653629 2.346371
     23 52 50.091017 1.908983
     24 16 15.940430 0.059570

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

count    62.000000
mean      1.754803
std       11.456846
min       -7.169714
25%       -0.515541
50%        0.424319
75%        1.869259
max       88.294950
Name: Diferencia, dtype: float64
```

```
+ Código + Texto RAM Disco Editando ^
```

```
[22] 23 52 50.091017 1.908983
      24 16 15.940430 0.059570

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

count    62.000000
mean      1.754803
std       11.456846
min       -7.169714
25%       -0.515541
50%        0.424319
75%        1.869259
max       88.294950
Name: Diferencia, dtype: float64

[24] from sklearn import metrics
import numpy as np

print("Raíz de la desviación media al cuadrado:", np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

Raíz de la desviación media al cuadrado: 11.498763363216353

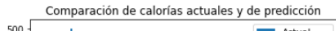
[26] import matplotlib.pyplot as plt

muestra_validacion.plot.bar(rot=0)

plt.title("Comparación de calorías actuales y de predicción")

plt.xlabel("Muestra de alimentos")
plt.ylabel("Cantidad de calorías")

plt.show()
```



```
[24] import numpy as np
      print("Raíz de la desviación media al cuadrado:", np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

Raíz de la desviación media al cuadrado: 11.498763363216353

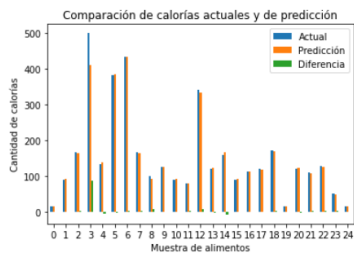
```
[26] import matplotlib.pyplot as plt

      muestra_validacion.plot.bar(rot=0)

      plt.title("Comparación de calorías actuales y de predicción")

      plt.xlabel("Muestra de alimentos")
      plt.ylabel("Cantidad de calorías")

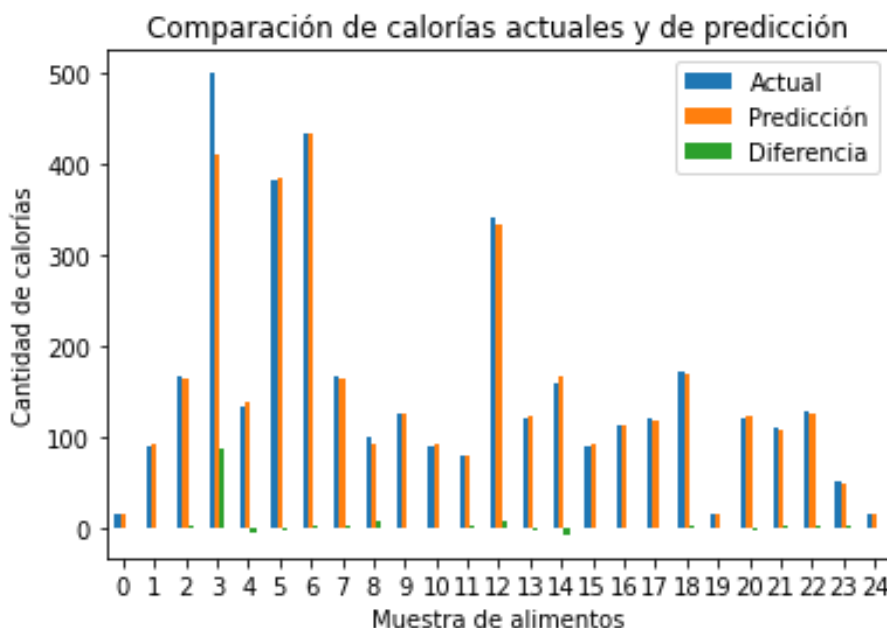
      plt.show()
```



Explicación del código

Primero importamos la librería `panda`, y le asignamos la variable `pd`, después abrimos el archivo que queremos, en este caso el Excel que contiene nuestros datos de las 14 semanas, y le asignamos la variable `datos_consumo`, usamos la función `head` para comprobar que nuestros datos se cargaron correctamente, usamos la función `groupby` para agrupar los datos y usamos `count` para obtener los subtotales, en otro apartado de código en el mismo archivo usamos la función `describe` para ver la estadística descriptiva de nuestro archivo, ahora iremos en otro apartado con la selección de datos, y seleccionamos las columnas de la 4 a la 7, usamos `datos_seleccionados` para que nuestro código se ejecute, ahora en otro apartado, pondremos la función `.info()` para ver la información de los datos del nuevo dataframe, y en este punto solo debemos tener valores numéricos, ahora iremos con la limpieza de datos en el mismo archivo, primero busquemos valores nulos utilizando `isnull().values.any()` en nuestros datos seleccionados, creamos un nuevo dataframe y le damos la variable de dataset, en seguida lo ejecutamos con `isnull().sum()` para validar que no tengamos valores nulos en ninguna columna, en

el modelo hecho por mí cumple las características, y en todas sale 0 y dando False de que no hay valores nulos, nuestra siguiente sección en el mismo código es preparar los datos y asignamos las variables “x” y “y”, x para variables independientes y y para variables dependientes, ahora dividimos nuestros datos para utilizar el 80% como entrenamiento y 20% como prueba para esto importamos la herramienta Sckit-Learn, ahora vamos con nuestra modelación de datos, en el que importamos la regresión lineal, y le asignamos la variable modelo_regresion, ahora utilizaremos fit para ajustar el modelo a nuestro conjunto de datos, en este momento ya el algoritmo sabe las variables x y y podemos verlo con x_columns = [carbohidratos,etc], ahora ya creado nuestro modelo probaremos nuestro modelo con valores de prueba y para esto utilizamos .predict, ya revisando la diferencia creamos un dataframe con los valores actuales y los de comparación y para esto elegimos una muestra de 25 valores, ahora usamos la función describe para obtener la estadística descriptiva de la columna “Diferencia”, importamos las métricas e imprimimos la raíz de la desviación media, ahora por ultimo validamos nuestros datos y para esto importamos la librería que nos permitirá gráficas, creamos un gráfico con el dataframe, indicamos el titulo con .title, asignamos a las x como muestra de alimentos con .xlabel y la y como la cantidad de calorías con .ylabel, por ultimo desplegamos nuestro gráfico



1.- ¿Cuántos intentos o corridas realizaste para obtener los resultados sin errores? Porqué.

Primero tuve que importar otro archivo con el mismo nombre con datos añadidos porque los datos que hice eran muy pocos, hice dos corridas, porque tenía datos nulos en mi archivo de Excel, y en otra función el código no se ejecutaba, así que lo tuve que correr otra vez.

2.- ¿Qué problemas se presentaron y cómo los resolviste?

Tuve algunos problemas con los datos, porque no los recuperaba y no encontraba mi cuenta, así que tuve que hacerlo desde mi celular, en las primeras fases tenía muchos errores pero creo que preguntarle a mi maestros fue de gran utilidad para poder entender mejor, también algunas veces no sabía, no me salían o no entendía las cosas, así que el hecho de preguntarle a mis profesores me ayudó mucho, también tuve que volver a ver las grabaciones en clase para ver cómo se iba a realizar las tareas encargadas, con el programa no tuve problemas que no se pudiesen solucionar, también tuve que corregir los errores que el profesor me marcaba en mis trabajos anteriores para no cometerlos en futuros trabajos. En este trabajo de fase cuatro, no se me presento problema alguno más que la recolección de datos, por parte de la programación no se presentó ningún problema.

3.- ¿Qué resultados arrojó el análisis? Explica los valores y la gráfica.

Primero tuve que ver que tan preciso era el modelo para proseguir con mis gráficas, al comprobar que $11.498 < 13.916$ vemos que el modelo que presentare es preciso, las columnas marcadas en azul son los valores actuales, los valores de el archivo que importamos anteriormente, las columnas marcadas en color naranja son los valores que nuestro algoritmo predijo con todos los datos dados, las columnas en color verde son la diferencia que hay entre esas dos, o mejor dicho el margen de error que hay entre las columnas actuales y las columnas de predicción.

4.- ¿Qué aprendiste y cuáles son tus conclusiones de la modelación?

Aprendí a programar, aprendí a que las matemáticas en la ciencia de datos nos ayudan a predecir modelos y esto se me hace interesante ya que lo podemos utilizar en la cotidianidad, el modelo utilizado fue sobre las calorías consumidas y si esto influye en la masa corporal para esto sacamos el consumo calórico en 14 semanas, estas predicciones del modelo hecho en el programa de Google Colab nos pueden ayudar en un futuro para combatir la obesidad presentado en el país de México, podemos concluir que nuestro modelo sirve para predecir precisamente, con datos confiables recolectados de una página confiable.

Efecto del consumo calórico en el tiempo

Código

```
import pandas as pd

datos_consumo = pd.read_excel('datos.xlsx')

datos = datos_consumo[["Fecha (dd/mm/aa)", "Calorías (kcal)"]]

datos.head()

suma_calorías = datos["Calorías (kcal)"].sum()

suma_calorías

días = datos["Fecha (dd/mm/aa)"].nunique()

días

calorías_promedio = suma_calorías/días

print("Tu promedio de calorías consumidas en", días, "días es:", calorías_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"):
    calorías_requeridas = 655+(9.56*peso)+(1.85*altura)-(4.68*edad)
elif(genero == "Hombre"):
    calorías_requeridas = 66.5+(13.75*peso)+(5*altura)-(6.8*edad)
print("Con base en tus datos, tu consumo de calorías al día debe ser de:", calorías_requeridas)

diferencia = calorías_promedio - calorías_requeridas

diferencia

efecto_anual = diferencia * 450/3500 * 365 /1000

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

```

```

[ ] import pandas as pd
    datos_consumo = pd.read_excel('datos.xlsx')

[ ] datos_consumo.head()

```

	Fecha (dd/mm/aa)	Momento	Nombre alimento	Calorías (kcal)	Carbohidratos (g)	Lípidos/grasas (g)	Proteína (g)	Sodio (mg)
0	2020-08-17	Desayuno	Kellogg's Cereal Extra Arándanos	113	25.00	0.50	2.00	110.0
1	2020-08-17	Desayuno	Lala Leche Deslactosada Light	91	12.00	1.30	7.80	116.0
2	2020-08-17	Comida	Arroz con pollo	299	38.01	10.88	11.32	544.0
3	2020-08-17	Cena	Tostadas con Frijoles, Queso, Carne, Lechuga, ...	151	12.54	7.85	8.12	274.0
4	2020-08-18	Desayuno	Kellogg's Cereal Extra Arándanos	113	25.00	0.50	2.00	110.0

```

[ ] datos = datos_consumo[["Fecha (dd/mm/aa)","Calorías (kcal)"]]
    datos.head()

```

	Fecha (dd/mm/aa)	Calorías (kcal)
0	2020-08-17	113
1	2020-08-17	91
2	2020-08-17	299
3	2020-08-17	151
4	2020-08-18	113

```

[ ] suma_calorias = datos["Calorías (kcal)"].sum()
    suma_calorias

```

17583

```
[ ] datos = datos_consumo[["Fecha (dd/mm/aa)","Calorías (kcal)"]]
datos.head()
```

	Fecha (dd/mm/aa)	Calorías (kcal)
0	2020-08-17	113
1	2020-08-17	91
2	2020-08-17	299
3	2020-08-17	151
4	2020-08-18	113

```
[ ] suma_calorias = datos["Calorías (kcal)"].sum()
suma_calorias

42583
```

```
[ ] dias = datos["Fecha (dd/mm/aa)"].nunique()
dias

50
```

```
[ ] calorías_promedio = suma_calorias/días
print("Tu promedio de calorías consumidas en", días,"días es:", calorías_promedio)

Tu promedio de calorías consumidas en 50 días es: 851.66
```

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

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

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

42583

```
[ ] dias = datos["Fecha (dd/mm/aa)"].nunique()
dias

50
```

```
[ ] calorías_promedio = suma_calorias/días
print("Tu promedio de calorías consumidas en", días,"días es:", calorías_promedio)

Tu promedio de calorías consumidas en 50 días es: 851.66
```

```
[ ] 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"):
    calorías_requeridas = 655+(9.56*peso)+(1.85*altura)-(4.68*edad)
elif(genero == "Hombre"):
    calorías_requeridas = 66.5+(13.75*peso)+(5*altura)-(6.8*edad)
print("Con base en tus datos, tu consumo de calorías al día debe ser de:", calorías_requeridas)

diferencia = calorías_promedio - calorías_requeridas

diferencia
```

```
Ingresa tu peso en kilogramos: 65
Ingresa tu altura en centímetros: 162
Ingresa tu edad en años: 18
Ingresa tu género, Mujer/Hombre: Mujer
Con base en tus datos, tu consumo de calorías al día debe ser de: 1401.8500000000001
```

```

Tu promedio de calorías consumidas en 50 días es: 851.66

[ ] 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"):
        calorías_requeridas = 655+(9.56*peso)+(1.85*altura)-(4.68*edad)
    elif(genero == "Hombre"):
        calorías_requeridas = 66.5+(13.75*peso)+(5*altura)-(6.8*edad)
    print("Con base en tus datos, tu consumo de calorías al día debe ser de:", calorías_requeridas)

    diferencia = calorías_promedio - calorías_requeridas

    diferencia

Ingresa tu peso en kilogramos: 65
Ingresa tu altura en centímetros: 162
Ingresa tu edad en años: 18
Ingresa tu género, Mujer/Hombre: Mujer
Con base en tus datos, tu consumo de calorías al día debe ser de: 1491.8600000000001
-640.2000000000002

efecto_anual = diferencia * 450/3500 * 365 /1000
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: -30.043671428571436 kg

```

Explicación del código

Primero usamos el comando panda para importar nuestros datos de nuestro archivo Excel llamado "datos", le asignamos la variable `datos_consumo` y empezamos a usar los demás comandos, primero usamos `.head` para imprimir los datos seleccionados, luego usamos `suma_calorias` en el que despliega el total de calorías consumidas durante las 11 semanas, después le asignamos la variable `días` a la función `nunique()` para que nos despliegue el total de los días únicos, después calculamos el promedio de calorías que se consumió y para esto vamos a hacer la operación `suma_calorias / días`, variables que ya se han asignado con anterioridad, imprimimos el promedio de calorías, después usamos `int(input(""))` para que el usuario, en este caso yo, ponga su peso, altura, edad y genero y teniendo estos datos usamos un ciclo `while` en el que se pondrá la ecuación de Harris-Benedict, if si es mujer, elif si es hombre y por ultimo imprimimos cual debería ser el consumo de calorías, específicamente para esa persona, en este caso yo, como penúltima parte asignamos la diferencia entre las calorías promedio

y las calorías requeridas, y para finalizar hacemos la operación del efecto anual y le asignamos la variable efecto anual, después imprimimos este resultado.

Reflexión final (Conclusiones):

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 se puede tener cambios en la masa corporal, en el programa hecho en Google Colab los resultados arrojaron que, si seguía consumiendo las calorías que consumo, bajare 30 kg en un año, las estimaciones son bastante precisas como se comprobó en la fase 4

¿Qué procedimiento para realizar una regresión lineal te pareció mejor, el realizado en Excel en la semana 4 o en Python en la semana 15? ¿Por qué?

Creo que las dos son buenas a su manera, en general me pareció bien la de Excel ya que es de un uso más sencillo y ya estaba familiarizada con el programa, mientras que en Python tuve que aprender desde cero y se complico un poco por la misma razón, los datos mostrados en los dos programas no cambian, solo tienes que hacer un uso diferente para poder analizarlos, Python me pareció mejor ya que solo tenia que importar el archivo y usar algunos comandos para desplegar toda la información, también creo que es mucho más sencillo cuando tienes errores analizarlos en Python, ya que usando un comando, puedes ver si tienes todos los registros en el archivo Excel llenados.

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

Es importante para mantener todo en orden, ya que la mayoría de las personas cuenta con un teléfono celular, computadora e inclusive una cuenta bancaria, para esto se tiene que hacer un uso correcto de la ciencia de datos, respetando la

privacidad e integridad de las personas, y también haciendo buen uso de sus conocimientos programando o creando modelos correctamente, la ética debe de estar implícita en la ciencia de datos empezando por los propósitos por el que quieras hacer uso de la ciencia de datos, luego tenemos la ética en la recolección de datos, en nuestro caso para este proyecto tuvimos que usar fuentes confiables, en otros casos se tiene que usar fuentes confiables y seguras, todo esto tiene como consecuencia una buena relación con el consumidor o cliente.

Bibliografía

B. (2019, 28 agosto). CRISP-DM: La metodología para poner orden en los proyectos. Sngular. <https://www.sngular.com/es/data-science-crisp-dm-metodologia/>

N. (2020, 7 octubre). Alimentación en Japón. NutriPharm. <https://www.nutripharmonline.com/alimentacion-en-japon/>

R. (2020b, agosto 21). Ciencia de datos, clave en el futuro de la información. De Luna Noticias. <https://www.deluna.com.mx/ciencia/ciencia-de-datos-clave-en-el-futuro-de-la-informacion/>