



Instituto Tecnológico y de Estudios Superiores de Monterrey

MA1042 Grupo 803

Matemáticas y Ciencia de Datos para la toma de decisiones

Profesor Germán Domínguez Solís

Evidencia 2. Proyecto de Ciencia de Datos.

Domingo 29 de noviembre del 2020

Sánchez Panduro Francisco Javier – A01639832

Introducción

La ciencia de datos es una ciencia que combina múltiples campos, como lo son la estadística, métodos científicos, y el análisis de datos, para extraer el valor de los datos. La ciencia de datos es importante porque permite a las organizaciones usar sus datos para mejorar sus productos, o mejorar la investigación e innovación (Oracle México, s.f.).

Además, de la ciencia de datos nace el aprendizaje automático, o Machine Learning, que permite que las computadoras extraigan los datos, y descubran cosas, y realicen aplicaciones de inteligencia artificial. Inteligencia artificial, siendo cuando se hace que una computadora imite un comportamiento humano (Oracle México, s.f.).

La ciencia de datos puede ayudar a las personas de muchas maneras, ya que se pueden analizar más datos de los que podría hacer cualquier persona, en cuestión de minutos. Si se desarrolla lo suficiente, se podrían planear y predecir partes del mundo que no se podía antes, y aplicarse para cosas como crear autos que eviten estar en accidentes, o planeación de la producción de comida, para ser más sustentable y producir lo suficiente para una región y su economía.

La intención de este proyecto, sin embargo, es menos ambiciosa, aunque sigue siendo importante. Se utilizarán mis datos alimenticios de 90 días del año, para poder hacer un modelo en Python que pueda predecir cuantas calorías tiene un alimento, dando la información de sus macronutrientes, cómo los son los lípidos, azúcares, y proteínas.

Además, se logrará observar si estos hábitos alimenticios son saludables, y si se continúan, cómo cambiaría mi peso a través del tiempo.

Fase 1. Entendimiento del negocio

¿Quién es el cliente?

En este caso, el análisis realizado será a partir de mi alimentación, por lo que yo mismo soy el cliente de mi propia evaluación de los datos.

¿Qué problemas estás tratando de resolver?

Se está buscando encontrar patrones de mala alimentación, y predecir cómo puede cambiar mi cuerpo si me sigo alimentando de esta manera. Es posible que la manera de alimentarme sea adecuada, y pueda predecir una mejora en mi salud, sin embargo, también es posible encontrar altos niveles de azúcares o grasas, que podrían causar enfermedades graves en partes más tardías de mi vida.

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

La solución es que encontrará patrones que yo no pueda ver a simple vista, y me dará datos estadísticos de mi alimentación y mis hábitos, para de esta manera formar unos mejores, o continuar con ellos si son ideales. La ciencia de datos simplemente me dará los datos, y yo con ellos sabré que tipo de cambio tengo que lograr en mi persona.

¿Qué necesitas aprender para poder desarrollar la solución o problemas?

Necesito saber cómo voy a analizar los datos. Antes que todo me tengo que preguntar, qué quiero hacer y lograr con estos datos. Después los tengo que recolectar y explorar. Estos datos siendo mi alimentación a lo largo del semestre. Después tengo que aprender la manera óptima de preparar estos datos para que sean de la mejor calidad posible.

Después tendré que aprender a modelarlos, diseñando pruebas y construyendo un modelo óptimo para lograr mi objetivo. Tendré que evaluar mi modelo y los datos que este me dé. Y finalmente con estos datos, una vez que tenga mis datos finales, tendré que aprender cómo interpretarlos, para saber si mi alimentación es adecuada.

¿Qué debes hacer para desarrollar tu solución?

Tendré que aprender más sobre estadística y ciencia de datos, y sobre alimentación, así como recopilar información de valor sobre mi alimentación. Después tendré que saber qué tipo de solución estoy buscando, para saber esto tengo que saber cual es la manera ideal de alimentarse. Y finalmente, tendré que conocer las habilidades prácticas que requeriré para hacer un correcto análisis de datos de mi alimentación.

Fase 2. Entendimiento de los datos

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

Los datos existentes son los datos de todo lo que he comido. Los datos adquiridos son los macronutrientes que he ingresado en la tabla, provenientes de fatsecret.com. Y todavía no se tienen datos adicionales.

¿Qué tipos de datos se analizarán?

Se analizarán datos numéricos.

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

La columna de calorías, ya que es la que tiene más variedad.

¿Qué atributos parecen irrelevantes y pueden ser excluidos?

El momento en el que fue ingerido, ya que es irrelevante para el análisis. Y la fuente, que, aunque es importante citar siempre las fuentes, y no plagiar información, se ha

utilizado la misma fuente para todos los macronutrientes, y es irrelevante para el análisis de datos.

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

No porque aun no termina el semestre, y mis hábitos alimenticios cambian constantemente, se podrían hacer conclusiones generales, pero es posible que exista un cambio grande al tener todos los datos.

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

No porque solo se utilizan los nutrientes más populares y relevantes, que se incluyen en todos los etiquetados, y que la gran mayoría de los alimentos los contienen. Si existieran datos como vitamina C, o cafeína, probablemente serían menos relevantes y crearían confusión.

¿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 alimentos que he consumido los he ingresado yo al comer al final del día o a veces al final de la semana. Los datos de los nutrientes se han obtenido de fatsecret.com.

Antes si presentaba diferentes fuentes de datos, sin embargo, eso hizo que mi modelo fuera menos confiable, es posible que diferentes fuentes midan diferentes alimentos con más o menos rigor, y por eso sean un poco diferentes. Al utilizar datos de las mismas fuentes, ha sido más confiable mi modelo.

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

Hasta ahora no parece que sea necesario utilizar más fuentes de datos, sería interesante, sin embargo, comparar los resultados utilizando diferentes fuentes, y ver cómo cambia el modelo, o si se compensarían los errores unos con otros.

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

Hay más de 150 alimentos diferentes registrados, cada uno que cumple con los 9 atributos. La calidad parece ser muy buena, ya que el modelo creado con Excel fue bastante confiable, y presentaba una alta confianza al predecir los nutrientes de otros alimentos.

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

Recordamos que la hipótesis presentada es: “¿Si consumo cierta cantidad calórica, puedo tener cambios en mi masa corporal (peso) en un determinado tiempo?”.

Los datos que estamos analizando son del semestre completo, y me he pesado cada mes, sin embargo, esos datos no se han presentado en la tabla, no obstante, se puede observar también que he aumentado de peso. De esta manera, los datos nos pueden mostrar la tendencia de qué tipo de alimentos he consumido cuándo, y si es que hay alguna relación entre la cantidad calórica que he consumido, y mi peso.

Fase 3. Preparación de los datos

¿Qué datos hay que seleccionar? Por qué.

Se seleccionan los datos numéricos, porque son aquellos con los que se puede hacer una regresión lineal, asimismo, se asignan a una variable dependiente o independiente, para de esta manera poder graficarlos más adelante.

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

Sí, se tienen que eliminar, ya que estos pueden interferir con el aprendizaje. Es por esto que se hace la limpieza de datos, estos datos son inútiles, y hacen nuestro modelo menos correcto. Al eliminarlos, nos aseguramos de que la calidad de nuestros datos será mejor.

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

Sí, pero se tendría que volver a hacer la limpieza. La mejor idea para agregar más datos, es crear otro script para limpiar los nuevos datos, y adjuntarlos, y volver a asignar las variables. Es complicado, y se tienen que repetir cosas que ya se han hecho, pero es posible.

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

En este caso, todos los datos son de la misma fuente, ya sea que se refiera a la fuente de información, toda fue consultada del mismo sitio, pero dentro del programa, sólo se utilizó una tabla, datos.xlsx, por ello, no fue necesario fusionar datos de diferentes fuentes, todo vino de esta tabla de excel, la cual realicé a lo largo del semestre con los alimentos que consumía.

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

Sí, dentro de la preparación de los datos, se ordenan para modelarlos, esto hace que sea más ágil el programa, y el aprendizaje sea más rápido. Sin embargo, si se ordenan antes de separar los conjuntos de entrenamiento y prueba, es posible que los datos de prueba sean muy diferentes a los de conjuntos. Siempre en estadística, por eso, es necesario que se seleccionen datos de manera aleatoria.

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

Sí, porque al final queremos saber si nuestro modelo es útil, por eso, entrenamos el modelo con una parte de nuestros datos, y luego utilizamos los datos separados en prueba para probar nuestro modelo. Los datos separados para entrenar nuestro modelo son diferentes a los de prueba, y de esta manera nos aseguramos que nuestro modelo funciona con datos diferentes a aquellos que se entraron al inicio.

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

Se organizaron los datos, se eliminaron los datos nulos, es decir, aquellas celdas donde no había nada, se eliminaron las cabeceras y el nombre de los alimentos, sólo se dejaron los valores números, para de esta manera lograr hacer ciencia de datos.

Fase 4. Modelación de los datos

Parte 1: Análisis de regresión en Python

En esta sección, vamos a hacer un análisis de regresión lineal con Python, que es un modelado simple de aprendizaje automático. Para lograr esto, se hizo el script que se muestra aquí abajo.

Lo primero que se hace es importar los datos del archivo de excel de datos con los alimentos consumidos que se van a analizar. Para esto se usa la librería pandas y los datos se asignan a la variable `datos_consumo`

Después, se seleccionan los datos que se van a utilizar, esto se hace asignando a la variable `datos_seleccionados`, los datos de las filas y columnas que se quieren.

Después pasamos a limpiar los datos, es decir, usamos diferentes métodos para buscar valores nulos, descartarlos, y después validar que este proceso se haya realizado correctamente.

Después se preparan los datos, es decir, asignamos los atributos a las variables. En la variable X se establecen los atributos de entrada, y en y los atributos de salida, es decir, aquellos que se van a calcular. Y dividimos nuestros datos en 80% de ellos para entrenar nuestro modelo, y el otro 20% para probarlo. Esto se hace con la librería de sklearn.

Ahora, con los datos ya en orden, pasamos a la modelación. Usamos de nuevo la librería sklearn, esta vez la clase de regresión lineal, y creamos un objeto llamado `modelo_regresión` de esta clase.

Usamos el método `fit()` para ajustar el modelo a nuestro conjunto de datos, y con esto, el algoritmo aprende cuáles son los coeficientes de X óptimos para satisfacer el modelo, y usamos otro código para desplegar los coeficientes y sus valores.

En seguida, pasamos a la parte de prueba, en la que usamos los demás datos para probar nuestro modelo de regresión, y vemos la diferencia con el pronóstico, y los datos que se tenían antes. Y usamos la raíz de la desviación media cuadrada para determinar qué tan preciso es el modelo, esperando que el valor sea menor al 10% de la media de y. En ese caso fue 4.27, por lo que podemos concluir que es un muy buen modelo.

Finalmente, pasamos a visualizar los datos, utilizamos la librería de matplotlib, y creamos una gráfica, establecemos los parámetros y los títulos y etiquetas, y obtenemos una gráfica que nos muestra la predicción comparado con los valores reales, y la diferencia. Podemos observar que se acerca mucho a los valores reales, demostrando que efectivamente una raíz de desviación de 4.27 es buena para nuestros propósitos.

Evidencia 2. Proyecto de Ciencia de Datos

Google Colaboratory interface showing a Jupyter Notebook with the following code and output:

```
[1] import pandas as pd
```

```
[2] datos_consumo = pd.read_excel('datos.xlsx')
```

```
datos_consumo.head()
```

| | Fecha (dd/mm/aa) | Momento | Nombre alimento | Calorías (kcal) | Lípidos/grasas (g) | Carbohidratos (g) | Proteína (g) | Sodio (mg) | Fuente |
|---|------------------|----------|--------------------|-----------------|--------------------|-------------------|--------------|------------|------------------|
| 0 | 2020-08-17 | Comida | Big Mac | 486 | 24.00 | 45.00 | 22.00 | 891.0 | fatsecret.com.mx |
| 1 | 2020-08-17 | Desayuno | Chocolate caliente | 190 | 5.47 | 29.90 | 8.80 | 95.0 | fatsecret.com.mx |
| 2 | 2020-08-17 | Desayuno | Chorizo | 283 | 24.80 | 2.42 | 11.66 | 719.0 | fatsecret.com.mx |
| 3 | 2020-08-17 | Desayuno | Huevo | 212 | 16.18 | 2.08 | 13.84 | 224.0 | fatsecret.com.mx |
| 4 | 2020-08-17 | Comida | McPatatas | 288 | 14.00 | 36.00 | 4.00 | 240.0 | fatsecret.com.mx |

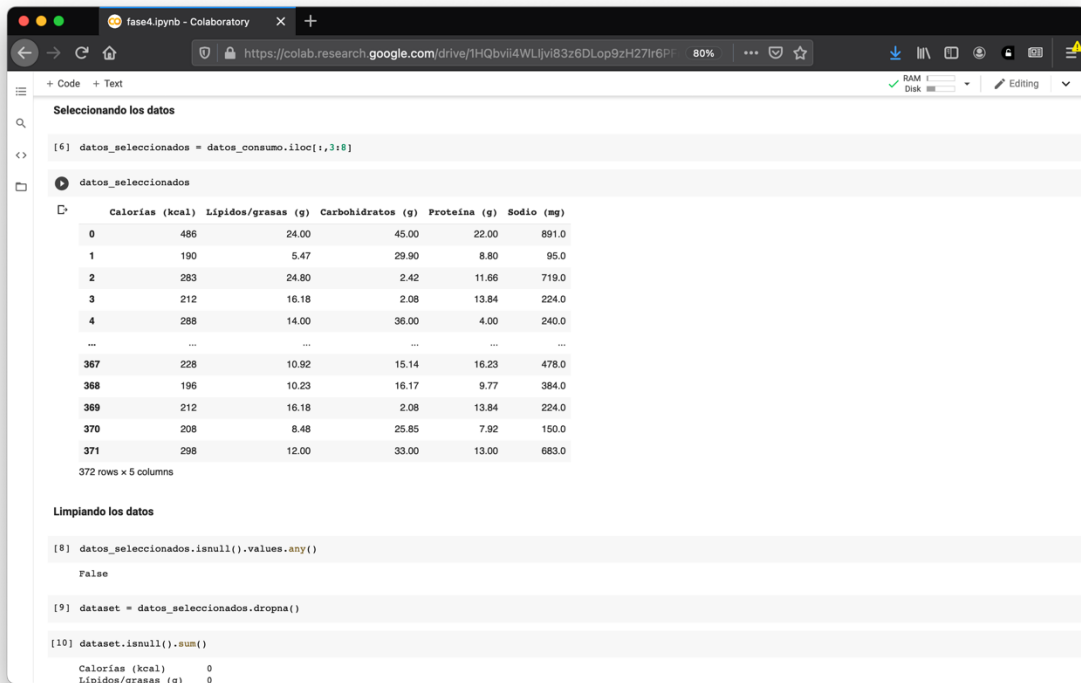
```
[4] datos_consumo.groupby("Momento").count()
```

| | Fecha (dd/mm/aa) | Nombre alimento | Calorías (kcal) | Lípidos/grasas (g) | Carbohidratos (g) | Proteína (g) | Sodio (mg) | Fuente |
|----------|------------------|-----------------|-----------------|--------------------|-------------------|--------------|------------|--------|
| Momento | | | | | | | | |
| Cena | 97 | 97 | 97 | 97 | 97 | 97 | 97 | 97 |
| Comida | 138 | 138 | 138 | 138 | 138 | 138 | 138 | 138 |
| Desayuno | 137 | 137 | 137 | 137 | 137 | 137 | 137 | 137 |

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

| | Calorías (kcal) | Lípidos/grasas (g) | Carbohidratos (g) | Proteína (g) | Sodio (mg) |
|-------|-----------------|--------------------|-------------------|--------------|------------|
| count | 372.000000 | 372.000000 | 372.000000 | 372.000000 | 372.000000 |
| mean | 195.790323 | 6.870349 | 24.674113 | 8.576048 | 311.185215 |
| std | 125.365489 | 6.338834 | 21.052010 | 7.352213 | 322.029722 |
| min | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| max | 486.000000 | 24.800000 | 45.000000 | 22.000000 | 891.000000 |

Evidencia 2. Proyecto de Ciencia de Datos



The screenshot shows a Google Colab notebook titled "fase4.ipynb - Colaboratory". The code is divided into two sections: "Seleccionando los datos" and "Limpiando los datos".

Seleccionando los datos

```
[6] datos_seleccionados = datos_consumo.iloc[1:318]
```

The output shows a DataFrame with 317 rows and 5 columns:

| | Calorias (kcal) | Lípidos/grasas (g) | Carbohidratos (g) | Proteína (g) | Sodio (mg) |
|-----|-----------------|--------------------|-------------------|--------------|------------|
| 0 | 486 | 24.00 | 45.00 | 22.00 | 891.0 |
| 1 | 190 | 5.47 | 29.90 | 8.80 | 95.0 |
| 2 | 283 | 24.80 | 2.42 | 11.66 | 719.0 |
| 3 | 212 | 16.18 | 2.08 | 13.84 | 224.0 |
| 4 | 288 | 14.00 | 36.00 | 4.00 | 240.0 |
| ... | ... | ... | ... | ... | ... |
| 367 | 228 | 10.92 | 15.14 | 16.23 | 478.0 |
| 368 | 196 | 10.23 | 16.17 | 9.77 | 384.0 |
| 369 | 212 | 16.18 | 2.08 | 13.84 | 224.0 |
| 370 | 208 | 8.48 | 25.85 | 7.92 | 150.0 |
| 371 | 298 | 12.00 | 33.00 | 13.00 | 683.0 |

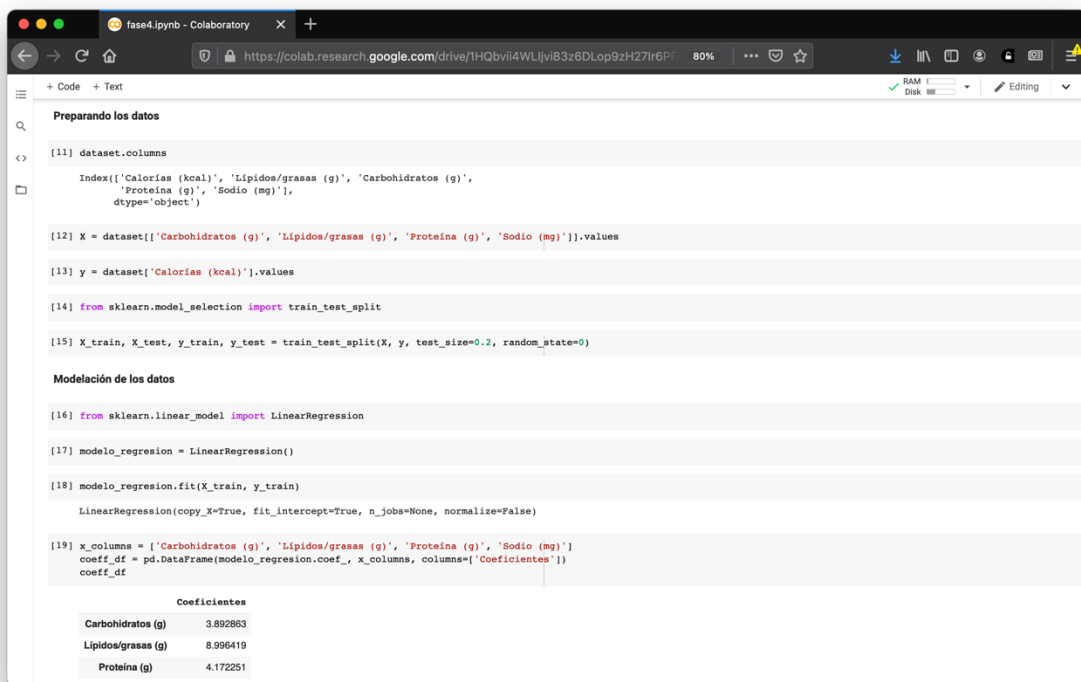
372 rows x 5 columns

Limpiando los datos

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

[9] dataset = datos_seleccionados.dropna()

[10] dataset.isnull().sum()
Calorias (kcal)    0
Lípidos/grasas (g)  0
```



The screenshot shows a Google Colab notebook titled "fase4.ipynb - Colaboratory". The code is divided into two sections: "Preparando los datos" and "Modelación de los datos".

Preparando los datos

```
[11] dataset.columns
Index(['Calorias (kcal)', 'Lípidos/grasas (g)', 'Carbohidratos (g)',
      'Proteína (g)', 'Sodio (mg)'],
      dtype='object')

[12] X = dataset[['Carbohidratos (g)', 'Lípidos/grasas (g)', 'Proteína (g)', 'Sodio (mg)']].values
[13] y = dataset['Calorias (kcal)'].values

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

Modelación de los datos

```
[16] from sklearn.linear_model import LinearRegression
[17] modelo_regresion = LinearRegression()
[18] 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
```

The output shows a DataFrame with 1 row and 4 columns:

| | Coeficientes |
|--------------------|--------------|
| Carbohidratos (g) | 3.892863 |
| Lípidos/grasas (g) | 8.996419 |
| Proteína (g) | 4.172251 |

Evidencia 2. Proyecto de Ciencia de Datos

```
fase4.ipynb - Colaboratory
https://colab.research.google.com/drive/1HQbvii4WLiJvI83z6DLop9zH27r6PF 80%
+ Code + Text
[20] y_pred = modelo_regresion.predict(X_test)

[21] validation = pd.DataFrame({'Actual': y_test, 'Predicción': y_pred, 'Diferencia': y_test-y_pred})

[22] muestra_validacion = validation.head(25)

[23] muestra_validacion
```

| | Actual | Predicción | Diferencia |
|----|--------|------------|------------|
| 0 | 69 | 69.139512 | -0.139512 |
| 1 | 17 | 19.054889 | -2.054889 |
| 2 | 45 | 42.158873 | 2.841127 |
| 3 | 439 | 437.102510 | 1.897490 |
| 4 | 0 | -0.439176 | 0.439176 |
| 5 | 439 | 437.102510 | 1.897490 |
| 6 | 105 | 115.966941 | -10.966941 |
| 7 | 322 | 319.990732 | 2.009268 |
| 8 | 213 | 212.753970 | 0.246030 |
| 9 | 121 | 121.523019 | -0.523019 |
| 10 | 190 | 193.812089 | -3.812089 |
| 11 | 493 | 498.344582 | -5.344582 |
| 12 | 439 | 437.102510 | 1.897490 |
| 13 | 23 | 23.885931 | -0.885931 |
| 14 | 84 | 84.980368 | -0.980368 |
| 15 | 486 | 491.727463 | -5.727463 |
| 16 | 322 | 319.990732 | 2.009268 |
| 17 | 162 | 155.300761 | 6.699239 |
| 18 | 49 | 52.736592 | -3.736592 |
| 19 | 123 | 124.167643 | -1.167643 |

```
fase4.ipynb - Colaboratory
https://colab.research.google.com/drive/1HQbvii4WLiJvI83z6DLop9zH27r6PF 80%
+ Code + Text
[24] validacion["Diferencia"].describe()

count    75.000000
mean     -1.089931
std       4.151573
min      -12.274690
25%      -3.305877
50%      -0.539494
75%       1.897490
max       6.699239
Name: Diferencia, dtype: float64

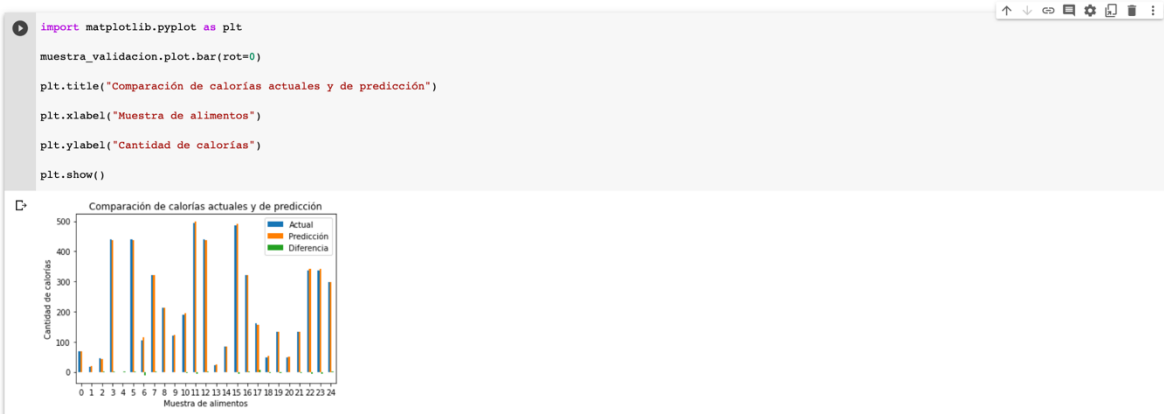
[25] from sklearn import metrics

[26] import numpy as np

[27] 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: 4.2654076252062145
```

Visualización de los datos



Parte 2: Modelación de los datos

¿Cuántos intentos o corridas realizaste para obtener los resultados sin errores?

Porqué

En este caso, el modelo funcionó con la primera corrida, esto se debe a muchos factores como la calidad de los datos, que mejoré y me aseguré de que fuera buena mientras los recolectaba, ya que al hacer el modelo en excel, se podía observar que la raíz media cuadrática era muy alta, y que el modelo no era confiable.

Asimismo, se ordenaron los datos y se limpiaron, lo cual nos dio solamente los datos numéricos, y que no hubieran cantidades que fueran a intrudir. Por esto, el entrenamiento se pudo hacer de mayor manera.

También se usaron librerías y hubo mucho código de ayuda, entonces esto hace que sea mucho más sencillo el trabajo que tener que codificar todas las funciones y clases desde cero.

¿Qué problemas se presentaron y cómo los resolviste?

Personalemente intenté instalar los paquetes para importar las librerías directamente en mi computadora y no tener que hacerlo desde Google Colab, ya que esta herramienta no te permite mantener los archivos en la nube. Pero esto resultó ser muy complicado, la última vez que trabajé con Numpy y Pandas y Matplotlib, tuve que desinstalar anaconda y Spyder, que fueron las herramientas a través de las cuales trabajé con ellos, para liberar espacio, y volver a instalarlo fue complicado.

Sin embargo, se nos brindó mucha ayuda para lograr nuestro objetivo, y no se necesitó usar más de un día seguido este código, por lo que utilizar Google Colab no fue tanto problema.

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

La raíz de la desviación media al cuadrado, también conocida como R^2 , fue de 4.2654, en general se espera que sea menor a 10 para ser confiable, siendo menor a 5 óptimo. Sin embargo, esto depende del uso y del caso, pero en esta circunstancia es un valor muy bueno.

La raíz de la desviación media al cuadrado es un valor en un análisis de regresión que nos dice que tanto se desvían los residuos en promedio del punto real, entonces, entre menor sea este número, esto significa que el valor pronosticado, se acerca bastante o es igual al valor real en muchas circunstancias.

A menos que haya una relación lineal perfecta entre dos datos, por ejemplo, la cantidad de dinero en pesos que tiene una persona, y cuantos dólares a un precio fijo puede comprar, este valor no será nunca cero, porque como podemos observar con el ejemplo, estos casos suelen ser muy difíciles de encontrar en el mundo real.

Después, la visualización de los datos nos da una mejor idea de cómo difieren los datos pronosticados con los reales. Las gráficas son buenas maneras de entender resultados abstractos, sin embargo, se suele perder precisión al utilizarlas, por ejemplo, en la gráfica resultante, podemos observar que el alimento 5 y el alimento 24 tienen una barra verde, que demuestra diferencia, de prácticamente el mismo color, sin embargo, podemos observar que la barra azul y la barra naranja, representando el valor actual y el valor pronosticado respectivamente, difiere más en el alimento 5 que en el 24.

Sin embargo, esta visualización nos ayuda para de una manera general, ver que nuestro modelo es bastante confiable si queremos predecir las calorías de un nuevo alimento.

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

La regresión lineal es una de las maneras más básicas para el manejo de datos, y esto es con buena razón. Es un método fácil de aprender, y fácil de aplicar, así como muy versátil. Sigue siendo uno de los métodos más populares para la ciencia de datos.

Además, con la programación, tareas más complicadas, como la reducción de dimensiones para generar la regresión lineal en dos planos, se vuelven tareas rápidas, que se pueden aplicar incluso si no se conoce exactamente lo que se está haciendo. Esto ayuda a abrir la ciencia de datos a más disciplinas, y a democratizarla, para que pueda ser utilizada por quien sea.

En mi opinión esto es algo maravilloso, ya que estas son herramientas poderosas y muy útiles, y el hecho de que otras disciplinas como medicina, ciencias políticas, e incluso los negocios las utilicen, mejoran la vida de todas las personas.

Fase 5: Efecto del consumo calórico en el tiempo

Para esta fase, se hará un script simple que calcule el total de calorías consumidas en el periodo, la cantidad de días, y con esto, se hará un promedio de las calorías consumidas al día en este período.

Lo primero que se hace es importar la librería pandas para poder leer el documento de excel, después este se asigna a la variable datos consumo, y se verifica que esté importada correctamente. Se limpian los datos para sólo utilizar las dos columnas que se necesitan, la de fecha y la de calorías. Se verifica que los datos se hayan asignado a la variable correctamente.

Después se empiezan a hacer los cálculos, el primero que se hace es el promedio de calorías al día. Después, con una ecuación pedimos el input del usuario, y usamos la ecuación de Harris-Benedict para determinar cuantas calorías debería de consumir esa persona al día. Luego se calcula la diferencia de las calorías que se deberían de consumir, y las calorías que se consumieron al día en promedio. Finalmente, se hace una predicción con estos datos, que determina cuál sería el peso que esta persona tendróa en una año si consumiera de esta manera.

Evidencia 2. Proyecto de Ciencia de Datos

The screenshot shows a Google Colab notebook titled "Evidencia 2. Proyecto de Ciencia de Datos". The browser address bar shows the URL: https://colab.research.google.com/drive/1Jp_VVzf8DF-iwZjU_L_F. The notebook interface includes a left sidebar with icons for file explorer, search, and code execution. The main area displays two code cells and their outputs.

Se importa librería pandas y se lee el archivo

```
[1] import pandas as pd
```

```
[2] #Se asignan los datos del excel a la variable datos_consumo
datos_consumo = pd.read_excel('datos.xlsx')

#Se comprueba que se hayan cargado correctamente
datos_consumo.head()
```

| | Fecha (dd/mm/aa) | Momento | Nombre alimento | Calorías (kcal) | Lípidos/grasas (g) | Carbohidratos (g) | Proteína (g) | Sodio (mg) | Fuente |
|---|------------------|----------|--------------------|-----------------|--------------------|-------------------|--------------|------------|------------------|
| 0 | 2020-08-17 | Comida | Big Mac | 486 | 24.00 | 45.00 | 22.00 | 891.0 | fatsecret.com.mx |
| 1 | 2020-08-17 | Desayuno | Chocolate caliente | 190 | 5.47 | 29.90 | 8.80 | 95.0 | fatsecret.com.mx |
| 2 | 2020-08-17 | Desayuno | Chorizo | 283 | 24.80 | 2.42 | 11.66 | 719.0 | fatsecret.com.mx |
| 3 | 2020-08-17 | Desayuno | Huevo | 212 | 16.18 | 2.08 | 13.84 | 224.0 | fatsecret.com.mx |
| 4 | 2020-08-17 | Comida | McPatatas | 288 | 14.00 | 36.00 | 4.00 | 240.0 | fatsecret.com.mx |

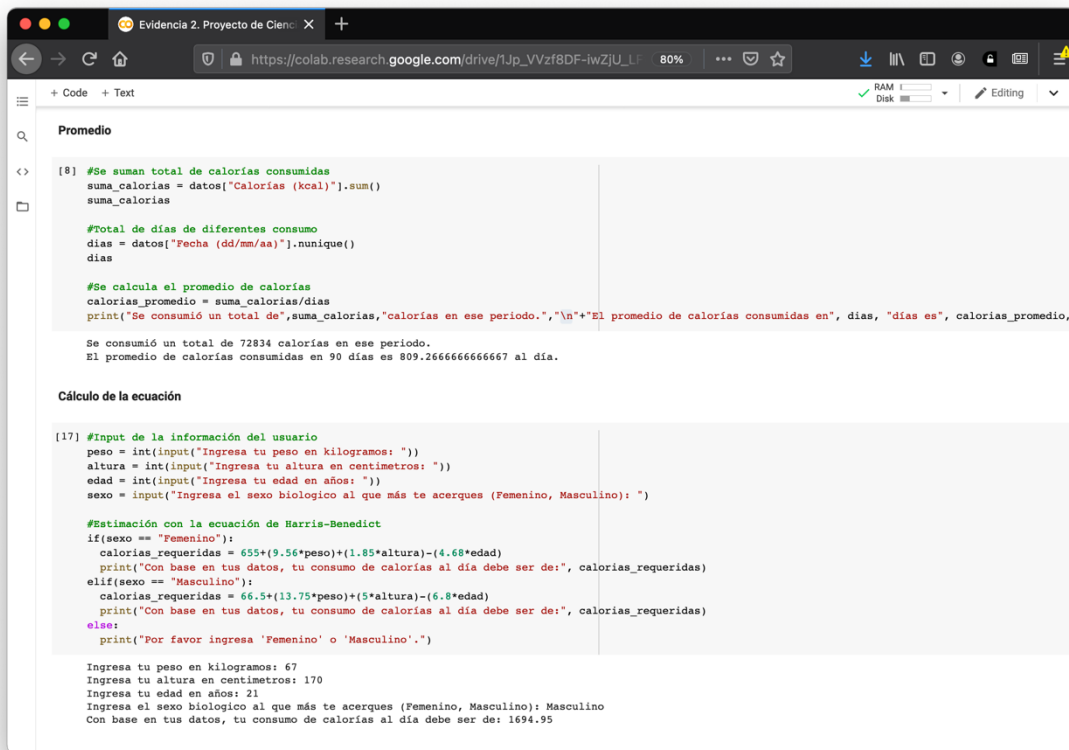
Limpeza de datos

```
[3] #Se asigna a datos solamente los datos que vamos a necesitar
datos = datos_consumo[["Fecha (dd/mm/aa)", "Calorías (kcal)"]]

#Se comprueba que se hayan cargado correctamente
datos.head()
```

| | Fecha (dd/mm/aa) | Calorías (kcal) |
|---|------------------|-----------------|
| 0 | 2020-08-17 | 486 |
| 1 | 2020-08-17 | 190 |
| 2 | 2020-08-17 | 283 |
| 3 | 2020-08-17 | 212 |
| 4 | 2020-08-17 | 288 |

Evidencia 2. Proyecto de Ciencia de Datos



The screenshot shows a Google Colab notebook with two code cells. The first cell, labeled [8], calculates the average calories consumed from a dataset. The second cell, labeled [17], takes user input for weight, height, age, and sex, then calculates the required calories using the Harris-Benedict equation.

```
[8] #Se suman total de calorías consumidas
suma_calorias = datos["Calorías (kcal)"].sum()
suma_calorias

#Total de días de diferentes consumo
dias = datos["Fecha (dd/mm/aa)"].nunique()
dias

#Se calcula el promedio de calorías
calorias_promedio = suma_calorias/dias
print("Se consumió un total de", suma_calorias, "calorías en ese periodo.", "\n" + "El promedio de calorías consumidas en", dias, "días es", calorias_promedio,

Se consumió un total de 72834 calorías en ese periodo.
El promedio de calorías consumidas en 90 días es 809.2666666666667 al día.

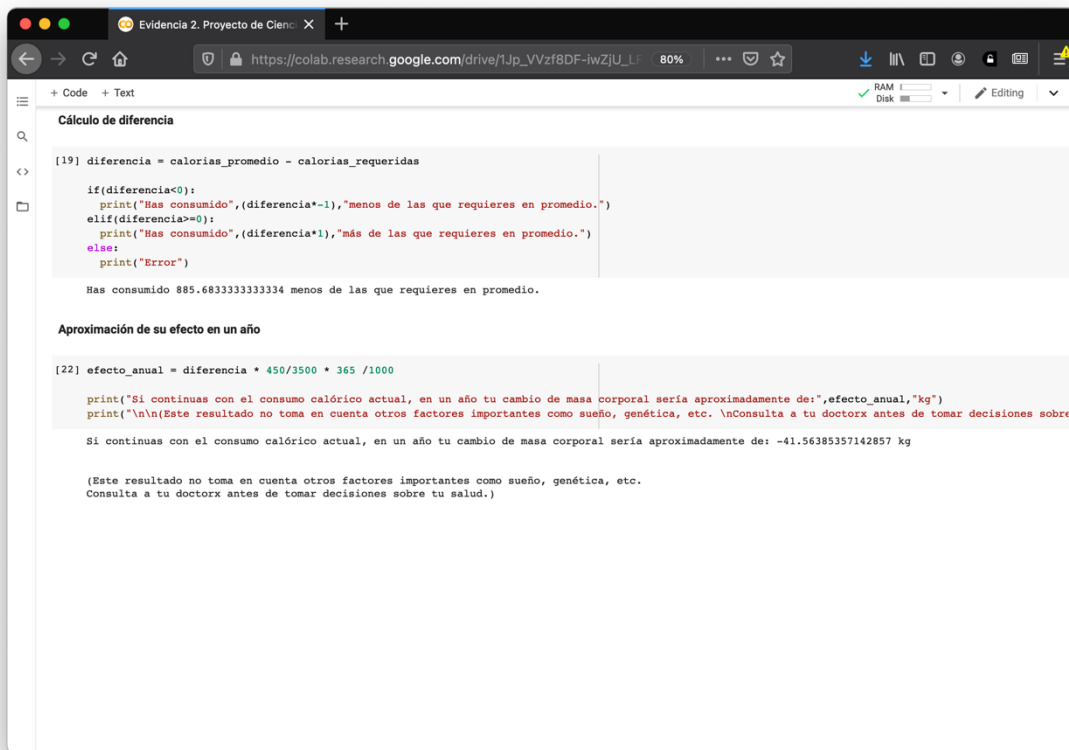
Cálculo de la ecuación

[17] #Input de la información del usuario
peso = int(input("Ingresa tu peso en kilogramos: "))
altura = int(input("Ingresa tu altura en centímetros: "))
edad = int(input("Ingresa tu edad en años: "))
sexo = input("Ingresa el sexo biológico al que más te acerques (Femenino, Masculino): ")

#Estimación con la ecuación de Harris-Benedict
if(sexo == "Femenino"):
    calorias_requeridas = 655+(9.56*peso)+(1.85*altura)-(4.68*edad)
    print("Con base en tus datos, tu consumo de calorías al día debe ser de:", calorias_requeridas)
elif(sexo == "Masculino"):
    calorias_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:", calorias_requeridas)
else:
    print("Por favor ingresa 'Femenino' o 'Masculino'.")

Ingresa tu peso en kilogramos: 67
Ingresa tu altura en centímetros: 170
Ingresa tu edad en años: 21
Ingresa el sexo biológico al que más te acerques (Femenino, Masculino): Masculino
Con base en tus datos, tu consumo de calorías al día debe ser de: 1694.95
```

Evidencia 2. Proyecto de Ciencia de Datos



```
[19] diferencia = calorias_promedio - calorias_requeridas

if(diferencia<0):
    print("Has consumido", (diferencia*-1), "menos de las que requieres en promedio.")
elif(diferencia==0):
    print("Has consumido", (diferencia*1), "más de las que requieres en promedio.")
else:
    print("Error")

Has consumido 885.6833333333334 menos de las que requieres en promedio.

Aproximación de su efecto en un año

[22] 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")
print("\n\n(Este resultado no toma en cuenta otros factores importantes como sueño, genética, etc. \nConsulta a tu doctorx antes de tomar decisiones sobre tu salud.)")

Si continúas con el consumo calórico actual, en un año tu cambio de masa corporal sería aproximadamente de: -41.56385357142857 kg

(Este resultado no toma en cuenta otros factores importantes como sueño, genética, etc.
Consulta a tu doctorx antes de tomar decisiones sobre tu salud.)
```

```
"""**Se importa librería pandas y se lee el archivo**
"""
```

```
import pandas as pd
```

```
#Se asignan los datos del excel a la variable datos_consumo
datos_consumo = pd.read_excel('datos.xlsx')
```

```
#Se comprueba que se hayan cargado correctamente
datos_consumo.head()
```

```
"""**Limpieza de datos**"""
```

```
#Se asigna a datos solamente los datos que vamos a necesitar
datos = datos_consumo[["Fecha (dd/mm/aa)", "Calorías (kcal)"]]
```

```
#Se comprueba que se hayan cargado correctamente
datos.head()

"""**Promedio**"""

#Se suman total de calorías consumidas
suma_calorias = datos["Calorías (kcal)"].sum()
suma_calorias

#Total de días de diferentes consumo
dias = datos["Fecha (dd/mm/aa)"].nunique()
dias

#Se calcula el promedio de calorías
calorias_promedio = suma_calorias/dias
print("Se consumió un total de",suma_calorias,"calorías en ese
periodo.", "\n"+"El promedio de calorías consumidas en", dias,
"días es", calorias_promedio, "al día." )

"""**Cálculo de la ecuación**"""

#Input de la información del usuario
peso = int(input("Ingresa tu peso en kilogramos: "))
altura = int(input("Ingresa tu altura en centímetros: "))
edad = int(input("Ingresa tu edad en años: "))
sexo = input("Ingresa el sexo biologico al que más te acerques
(Femenino, Masculino): ")

#Estimación con la ecuación de Harris-Benedict
if(sexo == "Femenino"):
```

Evidencia 2. Proyecto de Ciencia de Datos

```
calorias_requeridas = 655+(9.56*peso)+(1.85*altura)-(4.68*edad)
print("Con base en tus datos, tu consumo de calorías al día debe ser de:", calorias_requeridas)
elif(sexo == "Masculino"):
    calorias_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:", calorias_requeridas)
else:
    print("Por favor ingresa 'Femenino' o 'Masculino'.")

"""**Cálculo de diferencia**"""

diferencia = calorias_promedio - calorias_requeridas

if(diferencia<0):
    print("Has consumido",(diferencia*-1),"menos de las que requieres en promedio.")
elif(diferencia>=0):
    print("Has consumido",(diferencia*1),"más de las que requieres en promedio.")
else:
    print("Error")

"""**Aproximación de su efecto en un año**"""

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")  
print("\n\n(Este resultado no toma en cuenta otros factores  
importantes como sueño, genética, etc. \nConsulta a tu doctorx  
antes de tomar decisiones sobre tu salud.)")
```

Conclusiones

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.

Podemos concluir que la hipótesis es correcta, podemos observar con los cálculos de la fase 5, cómo podría cambiar el peso de una persona dependiendo de la cantidad calórica que consuma. También es importante tomar en cuenta otras cosas, por ejemplo, el nivel de actividad de esa persona. Si es una persona muy inactiva, necesitaría comer menos calorías que una persona muy activa, para mantener la misma masa corporal.

Sin embargo, existe un error en los cálculos, y este viene desde los datos. Para lograr mejores predicciones al momento de calcular las calorías de un alimento, las porciones de los alimentos consumidos no fueron tomadas en cuenta al ingresar los datos. Por eso, si se hubiera hecho una columna de datos en la que se especifique cuantas porciones del alimento ingresado se consumieron, se pudiera saber con exactitud las calorías promedio consumidas al día por el usuario. Es por esto, que al momento de calcular la diferencia entre calorías que se deberían de consumir, y las consumidas en promedio, se tiene un déficit tan grande.

Cual es el mejor procedimiento para realizar una regresión lineal

El procedimiento en Excel es mucho más sencillo porque se tiene una interfaz de usuario, y no es necesario saber exactamente lo que se está haciendo para conseguir todos los cálculos. Sin embargo, el programa es más lento que correr solamente un script, lo cuál podría ser un impedimento si se está intentando analizar una gran cantidad de datos, como es el caso con el Big Data.

Además, al utilizar Python se pueden hacer más acciones que no son posibles en excel, o que sería más tedioso hacerlo, como es trabajar con más dimensiones de datos y con matrices. Numpy es una herramienta muy poderosa que puede hacer esto por nosotros, así como librerías como “sklearn”, que nos permite hacer Machine Learning aún más avanzado.

También, leer las formulas en Python es más amigable, ya que en excel se tiene que saber la coordenada de la celda, y en Python se pueden nombrar variables para cada dato. Esto es útil por ejemplo, al momento de hacer la formula de Harris-Benedict, podemos simplemente ingresar “edad”, y no tener que buscar la coordenada E10, por ejemplo, y si se presenta un error, se sabe exactamente dónde se encuentra.

La ciencia de datos y la ética

Los datos son muy poderosos, sin embargo, es siempre importante saber qué se está tratando y cómo. Un ejemplo, es con los síntomas de COVID. Una gran parte de la población se infecta y es asintomática, y por ello, no se da cuenta de que está infectada, y suele infectar a más personas. Un programa que no tome esto en cuenta, podría preguntar solamente por los síntomas, y con eso hacer un diagnóstico, pero esto sería erróneo. Un mejor programa, podría usar aún más datos, como las ubicaciones recientes de esta persona, su trabajo, su edad, dónde reside, y con esto hacer un análisis más detallado. En Corea del Sur, se ha usado la ciencia de datos de esta manera para contener el virus mucho mejor que otros países (Schneider, Thornell, Parvaneh, Chakraborty, 2020).

Sin embargo, también es importante el consentimiento de la persona y sus datos, y aquí se vuelve complicado, porque si a no muchas personas les agrada dar sus datos, el modelo se puede volver menos confiable, y contraproducente.

Existen también alternativas, por ejemplo el API de Google y Apple, que permite a los gobiernos crear aplicaciones que rastreen de manera anónima las personas con quienes el usuario ha tenido contacto, y en caso de que alguna persona dé positivo a una prueba, lograr saber a quienes pudo haber contagiado, y prevenir que estas personas contagien a más personas (Google, s.f.).

Asimismo, no existen suficientes investigaciones que comprueben que estos programas realmente puedan detener el virus, y que esto no sea contraproducente para combatir la enfermedad por la velocidad por la que se propaga (Newton, 2020). Asimismo, es complicado para alguien que no sabe qué es un API, o que no sabe leer el código abierto proporcionado, que somos la gran mayoría de la población, entender si nuestra privacidad está segura en algo desarrollado por Google, la compañía publicitaria más grande del mundo, que usa la información personal de sus usuarios para proporcionar anuncios (Google, s.f.), y Apple, una compañía que dice estar a favor de la privacidad, pero ha tenido escándalos como en 2019, en la que se descubrió que las grabaciones de Siri estaban siendo escuchadas por personas para mejorar su algoritmo (Bogost, 2019).

Esta es una cuestión de ética con la ciencia de datos, cómo hay muchas, mi posición personal es que aunque es posible que los datos brinden una gran mejora en la calidad de vida de las personas, las personas siempre deben estar en control de sus datos, y si no quieren, por ejemplo, dar el historial de sus ubicaciones, aunque sea de manera anónima, se les debe respetar su deseo.

Referencias

Bogost Ian., 31 de enero 2019 *Apple's Empty Grandstanding About Privacy*. The Atlantic. Consultado el 29 de noviembre 2020 en:
<https://www.theatlantic.com/technology/archive/2019/01/apples-hypocritical-defense-data-privacy/581680/>

Google, s.f. *Exposure Notifications: Using technology to help public health authorities fight COVID-19*. Consultado el 29 de noviembre 2020 en: <https://www.google.com/covid19/exposurenotifications/>

Google, s.f. *Privacy & Terms*. Consultado el 29 de noviembre 2020 en: <https://policies.google.com/technologies/partner-sites?hl=en-US>

Newton C., 11 de abril 2020, *How Apple and Google are tackling one of the toughest parts about tracking COVID-19 exposures*. The Verge. Consultado el 29 de noviembre 2020 en: <https://www.theverge.com/interface/2020/4/11/21216652/apple-google-contact-tracing-covid-19-coronavirus-api-public-health-app-challenges>

Oracle México, s.f. *Ciencia de datos*. Consultado el 29 de noviembre 2020 en: <https://www.oracle.com/mx/data-science/what-is-data-science.html>

Schneider M., Thornell C., Parvaneh D., Chakraborty R., 6 de agosto 2020, *The big lesson from South Korea's coronavirus response*. Vox. Consultado el 29 de noviembre 2020 en: <https://www.vox.com/videos/2020/8/6/21356265/south-korea-coronavirus-response-testing>