Bellabeat ¿Cómo puede hacer una empresa tecnología para el bienestar para tomar decisiones inteligentes?

Jesús Emmanuel Román Acosta

- Tarea empresarial: Identificar tendencias de uso de los dispositivos inteligentes que sean aplicables a los clientes de Bellabeat para orientar futuras estrategias de marketing.

-Descripción de bases de datos utilizada: Datos de seguimiento de actividad física de Fitbit (CC0: Dominio público, conjunto de datos disponibles a través de Mobius): Este conjunto de datos de Kaggle contiene el seguimiento de la actividad física personal en treinta usuarios de Fitbit. Treinta usuarios elegibles de Fitbit prestaron su consentimiento para el envío de datos personales de seguimiento que incluyen rendimiento de la actividad física en minutos, ritmo cardíaco y monitoreo del sueño. Incluye información sobre la actividad diaria, pasos y ritmo cardíaco que se puede usar para explorar los hábitos de los usuarios.

Los datos, al sólo incluir información de 30 mujeres, no son del todo confiables ni suficientes para obtener ideas concluyentes. Asimismo, no poseen datos importantes como pesos, edad, etc.

-Documentación de limpieza y trasformación de datos:

Antes de comenzar el análisis estadístico de los datos, fue necesario corroborar la limpieza de los datos y trasformar algo de información. Utilizamos librerías de Python para este proceso.

-

```
#import libraries we'll be using
import numpy as np # data arrays
import pandas as pd # data structure and data analysis
import matplotlib as plt # data visualization
import datetime as dt # date time
```

```
#read the csv file an preview it
daily_activiy = pd.read_csv(r'C:/Users/jera2/OneDrive/Escritorio/CasoPractico2/database/dailyActivity_merged.csv')
daily_activiy.head(10)
```

	Id	ActivityDate	TotalSteps	TotalDistance	TrackerDistance	LoggedActivitiesDistance	VeryActiveDistance	ModeratelyActiveDistance	LightActiveDista
0	1503960366	4/12/2016	13162	8.50	8.50	0.0	1.88	0.55	
1	1503960366	4/13/2016	10735	6.97	6.97	0.0	1.57	0.69	
2	1503960366	4/14/2016	10460	6.74	6.74	0.0	2.44	0.40	
3	1503960366	4/15/2016	9762	6.28	6.28	0.0	2.14	1.26	
4	1503960366	4/16/2016	12669	8.16	8.16	0.0	2.71	0.41	
5	1503960366	4/17/2016	9705	6.48	6.48	0.0	3.19	0.78	
6	1503960366	4/18/2016	13019	8.59	8.59	0.0	3.25	0.64	
7	1503960366	4/19/2016	15506	9.88	9.88	0.0	3.53	1.32	
8	1503960366	4/20/2016	10544	6.68	6.68	0.0	1.96	0.48	
9	1503960366	4/21/2016	9819	6.34	6.34	0.0	1.34	0.35	

Luego, checamos si había NA:

```
#Lets check Nulls
nulls = daily_activiy.isnull().sum()
nulls
```

```
Id                          0
ActivityDate                0
TotalSteps                  0
TotalDistance               0
TrackerDistance             0
LoggedActivitiesDistance    0
VeryActiveDistance          0
ModeratelyActiveDistance    0
LightActiveDistance         0
SedentaryActiveDistance     0
VeryActiveMinutes           0
FairlyActiveMinutes         0
LightlyActiveMinutes        0
SedentaryMinutes            0
Calories                    0
dtype: int64
```

Ahora, checamos el tipo de dato de cada columna:

```
#Now lets check the data type to see if it makes sense
daily_activiy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 940 entries, 0 to 939
Data columns (total 15 columns):
Id                          940 non-null int64
ActivityDate                940 non-null object
TotalSteps                  940 non-null int64
TotalDistance               940 non-null float64
TrackerDistance             940 non-null float64
LoggedActivitiesDistance    940 non-null float64
VeryActiveDistance          940 non-null float64
ModeratelyActiveDistance    940 non-null float64
LightActiveDistance         940 non-null float64
SedentaryActiveDistance     940 non-null float64
VeryActiveMinutes           940 non-null int64
FairlyActiveMinutes         940 non-null int64
LightlyActiveMinutes        940 non-null int64
SedentaryMinutes            940 non-null int64
Calories                    940 non-null int64
dtypes: float64(7), int64(7), object(1)
memory usage: 110.2+ KB
```

Notamos que debemos cambiar el tipo de dato del campo ActivityDate de tipo objet a formato fecha con el comando:

```
#Lets convert activitydate type to datatime64 format mm/dd/yyyy
daily_activiy['ActivityDate'] = pd.to_datetime(daily_activiy['ActivityDate'], format="%m/%d/%Y")
daily_activiy.head()
```

	Id	ActivityDate	TotalSteps	TotalDistance	TrackerDistance	LoggedActivitiesDistance	VeryActiveDistance	ModeratelyActiveDistance	LightActiveDista
0	1503960366	2016-04-12	13162	8.50	8.50	0.0	1.88	0.55	
1	1503960366	2016-04-13	10735	6.97	6.97	0.0	1.57	0.69	
2	1503960366	2016-04-14	10460	6.74	6.74	0.0	2.44	0.40	
3	1503960366	2016-04-15	9762	6.28	6.28	0.0	2.14	1.26	
4	1503960366	2016-04-16	12669	8.16	8.16	0.0	2.71	0.41	

Luego, agregamos 3 nuevas columnas que servirán para nuestro análisis: día de la semana, total de minutos de ejercicio y total de horas:

```
#I'll add some new columns called weekday, total minutes and hours in order to use them in future analysis
new_col = ['Id', 'date', 'WeekDay', 'TotalSteps', 'TotalDistance', 'TrackerDistance', 'LoggedActivitiesDistance', 'VeryActiveDis

data = daily_activiy.reindex(columns=new_col)

data.head()
```

	Id	date	WeekDay	TotalSteps	TotalDistance	TrackerDistance	LoggedActivitiesDistance	VeryActiveDistance	ModeratelyActiveDistance	LightActivel
0	1503960366	2016-04-12	NaN	13162	8.50	8.50	0.0	1.88	0.55	
1	1503960366	2016-04-13	NaN	10735	6.97	6.97	0.0	1.57	0.69	
2	1503960366	2016-04-14	NaN	10460	6.74	6.74	0.0	2.44	0.40	
3	1503960366	2016-04-15	NaN	9762	6.28	6.28	0.0	2.14	1.26	
4	1503960366	2016-04-16	NaN	12669	8.16	8.16	0.0	2.71	0.41	

Llenamos la columna de día de la semana:

```
data['WeekDay']=data['date'].dt.day_name()
data.head(5)
```

	Id	date	WeekDay	TotalSteps	TotalDistance	TrackerDistance	LoggedActivitiesDistance	VeryActiveDistance	ModeratelyActiveDistance	LightActiv
0	1503960366	2016-04-12	Tuesday	13162	8.50	8.50	0.0	1.88	0.55	
1	1503960366	2016-04-13	Wednesday	10735	6.97	6.97	0.0	1.57	0.69	
2	1503960366	2016-04-14	Thursday	10460	6.74	6.74	0.0	2.44	0.40	
3	1503960366	2016-04-15	Friday	9762	6.28	6.28	0.0	2.14	1.26	
4	1503960366	2016-04-16	Saturday	12669	8.16	8.16	0.0	2.71	0.41	

Ahora, sumamos los minutos totales de ejercicio (muy activo, moderado, ligero) para llenar la columna minutos totales. De esta, obtenemos las horas dividiendo por 60:

We'll add all the column minutes to treat them as a whole, without considering sedentary time spent

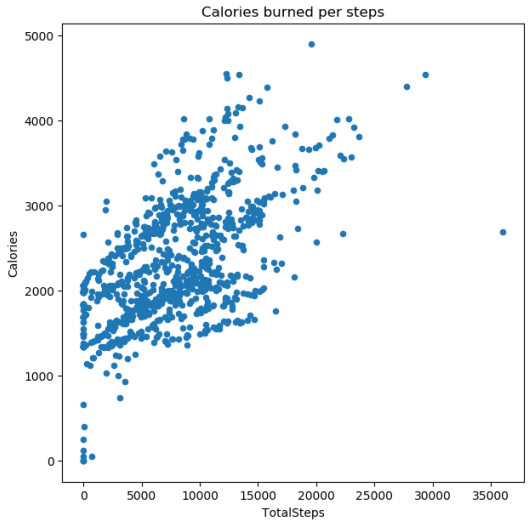
```
data['TotalMinutes']=data["VeryActiveMinutes"] + data["FairlyActiveMinutes"] + data["LightlyActiveMinutes"]
data['TotalMinutes'].head()
```

```
0    366
1    257
2    222
3    272
4    267
Name: TotalMinutes, dtype: int64
```

```
#Now, I'll convert them to hours
data['TotalMinutes']/60
data['TotalHours']=data['TotalMinutes']/60
```

- Análisis estadístico de los datos:

Con los datos listos, procedemos a formar algunas gráficas para tratar de encontrar relaciones entre las variables. Primero, graficamos las calorías en función de los pasos dados:



Notamos que las calorías tienen una correlación positiva con los pasos registrados. Crece su cantidad aceleradamente de 0 a 14,000 pasos, y después disminuye su rapidez de crecimiento. Hecho que se corrobora por sus valores de correlación:

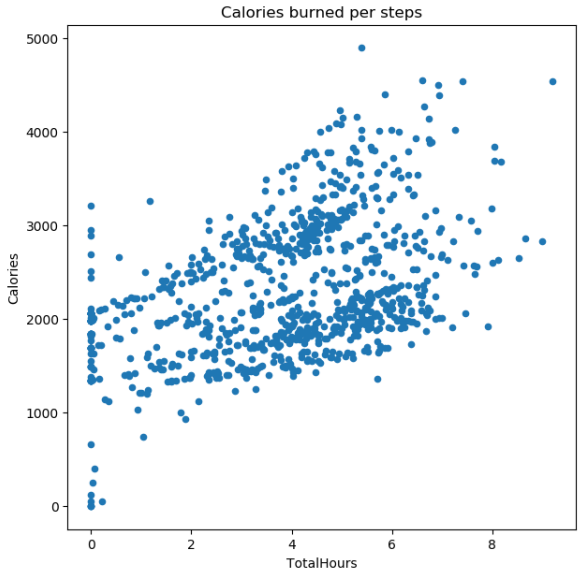
```
#Lets check correlation value
data[['TotalSteps','Calories']].corr()
```

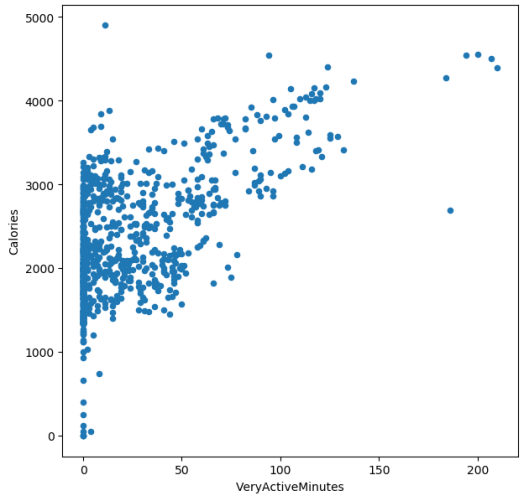
	TotalSteps	Calories
TotalSteps	1.000000	0.591568
Calories	0.591568	1.000000

```
#We could verify this fact taking just more than 15,000 totalsteps and see what happens to correlation value
copy = data[['TotalSteps','Calories']]
copy[copy.TotalSteps > 15000].corr()
```

	TotalSteps	Calories
TotalSteps	1.000000	0.345173
Calories	0.345173	1.000000

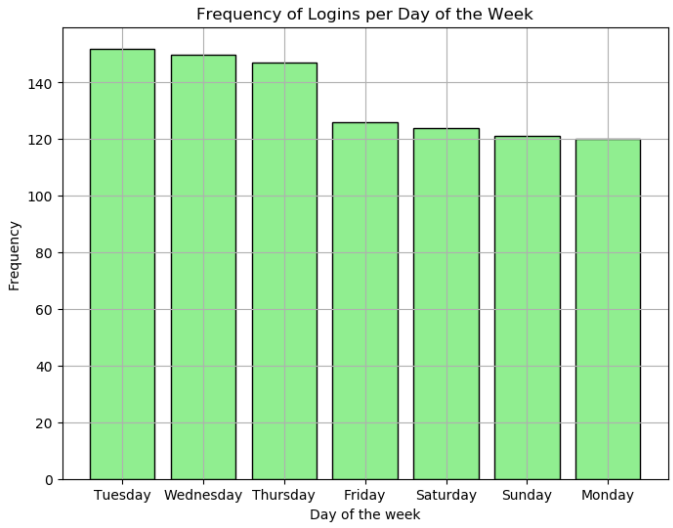
Este hecho se ve también reflejado por:





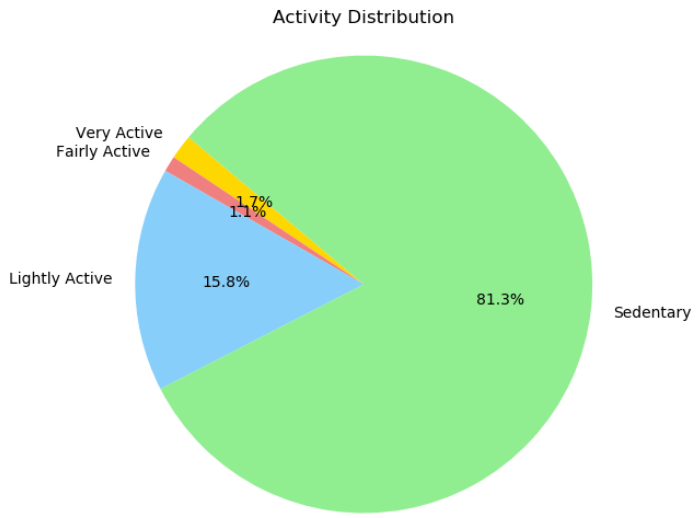
Notamos que existe una correlación considerable entre el tiempo invertido en ejercicio intenso y las calorías quemadas.

Luego, analizamos la distribución semanal de la frecuencia de uso del servicio. Esto es, que tanto utilizan la app cada día:



Notamos que su uso es mayor entresemana (martes-jueves) y disminuye en fines de semana y lunes.

Finalmente, analizamos las proporciones de usos del servicio:



Esto es, la mayor parte del tiempo no se utiliza la app para registrar actividad física. Siendo que el 82% del tiempo permanece inactivo.

- Conclusiones:

1. Los usuarios utilizan el servicio mayormente entresemana. Podría ser atractivo buscar una forma de incentivar al cliente a utilizarlo en fines de semana por medio de retos, notificaciones, etc.

2. La mayor parte del tiempo registrado en el servicio es ocio. Así, resultaría interesante generar alguna campaña que motive al usuario a aumentar su tiempo activo y registrarlo en la app.

3. Generar campañas que informen al cliente sobre las ventajas de realizar actividad deportiva diariamente. Esto se ve reflejado por el aumento drástico de las calorías quemadas al caminar más. En ese contexto, mantener al usuario detalladamente informado sobre sus pasos dados y formas de tener hábitos sanos sería un área de oportunidad.

4. Es importante comentar que, debido a la limitada cantidad de muestra, no es totalmente concluyente lo que se obtuvo en este reporte. Sería oportuno tener información de una cantidad mayor de usuarios. Así, habría que generar nuevas formas de recabar datos más completos que incluya información como el peso, edad, etc.