REPORTE DE TEXTO CON LAS MÉTRICAS DEL TITANIC

INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

JULIO 2023

Profesor Asignado: Alan Badillo Salas

Elaboró: *Miguel Angel Galindo Torres*

Correo: *angel.galindo.torres@hotmail.com*

Introducción

Dada la información disponible sobre el Titanic, se necesita conocer las probabilidades de ciertos eventos, la información se encuentra en un archivo csv, a partir de la información que este archivo puede proporcionar se utiliza python y la librería pandas para realizar una serie de cálculos y finalmente, utilizando estos cálculos, obtener las probabilidades de los eventos que son de interés para este reporte, como la probilidad de sobrevivir dado que se es mujer, entre algunos otros que se verán más adelante. La información del archivo se descarga en un DataFrame y a partir de este se llevan a acabo una serie de filtros para conocer algunos totales como supervivientes hombres o mujeres, entre otros distintos. Para la finalidad anterior, se crean un par de funciones, una de ellas, regresa la cantidad de filas filtradas (de una sola columna) bajo ciertas condiciones y la segunda tiene una función similar pero utilizando dos columnas.

Justificación

Para contestar preguntas tales como la cantidad de hombres o mujeres sobrevivientes, cuantos adultos o menores de edad murieron y si la clase a la que pertenece un pasajero incide en la probabilidad de haber sobrevivido o no, es que se crea este reporte. Además, teniendo en cuenta que este es un curso en el que se aprende sobre la manipulación de información con ayuda de python y algunas librerías como Pandas que ayudan a este fin ,es una oportunidad para practicar algunas de los temas aprendidos hasta el momento en el curso Programación Python en el Ámbito Científico.

Pasos a resolver

Hay una serie de preguntas que se necesitan responder a partir de la información disponible, el set de preguntas es el siguiente:

- Total de supervivientes
- Total de supervivientes mujeres
- Total de supervivientes hombres
- Total de supervivientes mayores a 18 años
- Total de supervivientes menores a 18 años
- Total de supervivientes mayores a 50 años
- Total de muertos
- Total de muertos mujeres
- Total de muertos hombres
- Tasa de supervivencia de la clase 1
- Tasa de supervivencia de la clase 2
- Tasa de supervivencia de la clase 3
- Tasa de supervivencia de la clase 1 siendo mujer
- Tasa de supervivencia de la clase 2 siendo mujer
- Tasa de supervivencia de la clase 3 siendo mujer
- Probabilidad de sobrevivir dado que se es mujer (usar regla de Bayes)
- Probabilidad de sobrevivir dado que se es hombre (usar regla de Bayes)
- Probabilidad de sobrevivir dado que se es mujer y está en la clase 1 (usar regla de Bayes compuesta)

El primer paso para contestar a la serie de preguntas planteadas en la justificación, ha sido la adquisición de la información sobre el Titanic que se ha obtenido del siguiente link:

https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv

El siguiente paso es cargar el contenido del archivo csv a un DataFrame de Pandas, como se muestra en la siguiente imagen.

[1]:
```
#1.- ADQUISICION DE DATOS

#Importamos las librerías pandas y numpy
import pandas as pd
import numpy as np

#Leemos la información del archivo titanic.cs y
#Obtenemos una muestra de 5 registros
info = pd.read_csv("https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv")
info.sample(5)
```

[1]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
19	20	1	3	Masselmani, Mrs. Fatima	female	NaN	0	0	2649	7.225	NaN	C
435	436	1	1	Carter, Miss. Lucile Polk	female	14.0	1	2	113760	120.000	B96 B98	S
585	586	1	1	Taussig, Miss. Ruth	female	18.0	0	2	110413	79.650	E68	S
184	185	1	3	Kink-Heilmann, Miss. Luise Gretchen	female	4.0	0	2	315153	22.025	NaN	S
387	388	1	2	Buss, Miss. Kate	female	36.0	0	0	27849	13.000	NaN	S

Una vez que tenemos cargada la información en un DataFrame, podemos

empezar a trabajar con ella, pero antes modificamos el nombre de las columnas para que estas sean más fáciles de comprender.

```
#2.- Modificando los nombres de las columnas
info.columns = ["ID", "SOBREVIVE", "CLASE", "NOMBRE", "SEXO", "EDAD",
                "HERMANOS", "PADRES", "TICKET", "TARIFA", "CABINA", "MUELLE"]

#Obtenemos une muestra de 5 registros
info.sample(5)
```

	ID	SOBREVIVE	CLASE	NOMBRE	SEXO	EDAD	HERMANOS	PADRES	TICKET	TARIFA	CABINA	MUELLE
630	631	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.0	0	0	27042	30.00	A23	S
505	506	0	1	Penasco y Castellana, Mr. Victor de Satode	male	18.0	1	0	PC 17758	108.90	C65	C
563	564	0	3	Simmons, Mr. John	male	NaN	0	0	SOTON/OQ 392082	8.05	NaN	S
626	627	0	2	Kirkland, Rev. Charles Leonard	male	57.0	0	0	219533	12.35	NaN	Q
529	530	0	2	Hocking, Mr. Richard George	male	23.0	2	1	29104	11.50	NaN	S

A continuación, se lleva a cabo la limpieza de datos, en este caso en particular, consta de 3 aspectos, el primero implica cmabiar los valores de la columna SEXO de female a MUJER y de male a HOMBRE

```
#3.- Limpieza de DATOS
#Mapeamos la columna SEXO (MUJER, HOMBRE)
info["SEXO"] = info["SEXO"].map({
    "female":"MUJER",
    "male":"HOMBRE"
})

#Obtenemos une muestra de 5 registros
info.sample(5)
```

	ID	SOBREVIVE	CLASE	NOMBRE	SEXO	EDAD	HERMANOS	PADRES	TICKET	TARIFA	CABINA	MUELLE
301	302	1	3	McCoy, Mr. Bernard	HOMBRE	NaN	2	0	367226	23.2500	NaN	Q
196	197	0	3	Mernagh, Mr. Robert	HOMBRE	NaN	0	0	368703	7.7500	NaN	Q
342	343	0	2	Collander, Mr. Erik Gustaf	HOMBRE	28.0	0	0	248740	13.0000	NaN	S
502	503	0	3	O'Sullivan, Miss. Bridget Mary	MUJER	NaN	0	0	330909	7.6292	NaN	Q
21	22	1	2	Beesley, Mr. Lawrence	HOMBRE	34.0	0	0	248698	13.0000	D56	S

Después, se modifican algunos valores de la columna EDAD, ya que existen registros que no cuentan con este dato, para calcular estas edades, se utiliza una función que genera números aleatorios.

```
# Modificamos columna EDAD ya que existen valores Nan
def generaEdades(edad):
    if np.isnan(edad):
        return np.random.normal(info["EDAD"].mean(),info["EDAD"].std())
    else:
        return edad

info["EDAD"] = info["EDAD"].map(generaEdades).map(int)

#Obtenemos une muestra de 5 registros
info.sample(5)
```

	ID	SOBREVIVE	CLASE	NOMBRE	SEXO	EDAD	HERMANOS	PADRES	TICKET	TARIFA	CABINA	MUELLE
631	632	0	3	Lundahl, Mr. Johan Svensson	HOMBRE	51	0	0	347743	7.0542	NaN	S
88	89	1	1	Fortune, Miss. Mabel Helen	MUJER	23	3	2	19950	263.0000	C23 C25 C27	S
867	868	0	1	Roebling, Mr. Washington Augustus II	HOMBRE	31	0	0	PC 17590	50.4958	A24	S
154	155	0	3	Olsen, Mr. Ole Martin	HOMBRE	36	0	0	Fa 265302	7.3125	NaN	S
335	336	0	3	Denkoff, Mr. Mitto	HOMBRE	9	0	0	349225	7.8958	NaN	S

En seguida, se modifican algunos datos de la columna CABINA, ya que contiene valores NAN

```
# Modificamos columna CABINA ya que existen valores Nan
info["CABINA"] = info["CABINA"].fillna("X")

#Obtenemos une muestra de 5 registros
info.sample(5)
```

	ID	SOBREVIVE	CLASE	NOMBRE	SEXO	EDAD	HERMANOS	PADRES	TICKET	TARIFA	CABINA	MUELLE
175	176	0	3	Klasen, Mr. Klas Albin	HOMBRE	18	1	1	350404	7.8542	X	S
794	795	0	3	Dantcheff, Mr. Ristiu	HOMBRE	25	0	0	349203	7.8958	X	S
110	111	0	1	Porter, Mr. Walter Chamberlain	HOMBRE	47	0	0	110465	52.0000	C110	S
645	646	1	1	Harper, Mr. Henry Sleeper	HOMBRE	48	1	0	PC 17572	76.7292	D33	C
192	193	1	3	Andersen-Jensen, Miss. Carla Christine Nielsine	MUJER	19	1	0	350046	7.8542	X	S

Para realizar el cálculo de la información estadística, se han construido 2 funciones. La primera función filtra el DataFrame original con respecto a algún valor específico de una sola columna, el valor devuelto de esta función puede ser la cantidad de filas que cumplen con la condición específica o puede devolver el DataFrame filtrado que cumplen de igual manera, con la condición específica

```
#4.- Se crea función para calcular el total de Filas Filtradas por una Columnna del DataFrame
# infoOriginal es el DataFrame Original
# columna      es la columna a filtrar
# operador     es el operador relacional a utilizar (>, <, =, >=, <=)
# condicion    es la condicióna a cumplir
# filas        es un bool, True devuelve el número de filas filtradas, False devuelve el DataFrame filtrado
def totales1Columna(infoOriginal, columna, operador, condicion, filas):
    if operador == ">":
        infoFiltrar = infoOriginal[columna] > condicion
    elif operador == "<":
        infoFiltrar = infoOriginal[columna] < condicion
    elif operador == "=":
        infoFiltrar = infoOriginal[columna] == condicion
    elif operador == ">=":
        infoFiltrar = infoOriginal[columna] >= condicion
    elif operador == "<=":
        infoFiltrar = infoOriginal[columna] <= condicion
    else:
        infoFiltrar = infoOriginal[columna] != condicion

    infoFiltrada = infoOriginal[infoFiltrar]

    if filas == True:
        return len(infoFiltrada)
    else:
        return infoFiltrada
```

La segunda función filtra el DataFrame original con respecto a algún valor específico de dos columna, el valor devuelto de esta función puede ser la cantidad de filas que cumplen con las condiciones específica o puede devolver el DataFrame filtrado que cumplen de igual manera, con las condiciones que se pasan como argumentos.

```
#5.- Se crea función para calcular el total de Filas Filtradas por dos Columnna del DataFrame
# infoOriginal es el DataFrame Original
# columna1     es la primera columna a filtrar
# operador1    es el primer operador relacional a utilizar (>, <, =, >=, <=) y sus combinaciones
# condicion1   es la primera condicióna a cumplir
# opBool       es el operador lógico a utilizar (and)
# columna2     es la segunda columna a filtrar
# operador2    es el segundo operador relacional a utilizar (>, <, =, >=, <=) y sus combinaciones
# condicion2   es la segunda condicióna a cumplir
# filas        es un bool, True devuelve el número de filas filtradas, False devuelve el DataFrame filtrado

def totales2Columnas(infoOriginal, columna1, operador1, condicion1, opBool, columna2, operador2
                     ,condicion2, filas):
    if operador1 == ">" and operador2 == ">" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] > condicion1) & (infoOriginal[columna2] > condicion2)
    elif operador1 == ">" and operador2 == "<" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] > condicion1) & (infoOriginal[columna2] < condicion2)
    elif operador1 == "<" and operador2 == "<" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] < condicion1) & (infoOriginal[columna2] < condicion2)
    elif operador1 == "<" and operador2 == ">" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] < condicion1) & (infoOriginal[columna2] > condicion2)
    elif operador1 == ">" and operador2 == ">=" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] > condicion1) & (infoOriginal[columna2] >= condicion2)
    elif operador1 == ">" and operador2 == "<=" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] > condicion1) & (infoOriginal[columna2] <= condicion2)
    elif operador1 == "<" and operador2 == "<=" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] < condicion1) & (infoOriginal[columna2] <= condicion2)
    elif operador1 == "<" and operador2 == ">=" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] < condicion1) & (infoOriginal[columna2] >= condicion2)
    elif operador1 == ">=" and operador2 == ">" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] >= condicion1) & (infoOriginal[columna2] > condicion2)
    elif operador1 == ">=" and operador2 == "<" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] >= condicion1) & (infoOriginal[columna2] < condicion2)
```

```
    elif operador1 == "<=" and operador2 == "<" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] <= condicion1) & (infoOriginal[columna2] < condicion2)
    elif operador1 == "<=" and operador2 == ">" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] <= condicion1) & (infoOriginal[columna2] > condicion2)
    elif operador1 == "=" and operador2 == "=" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] == condicion1) & (infoOriginal[columna2] == condicion2)
    elif operador1 == "=" and operador2 == ">=" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] == condicion1) & (infoOriginal[columna2] >= condicion2)
    elif operador1 == "=" and operador2 == "<=" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] == condicion1) & (infoOriginal[columna2] <= condicion2)
    elif operador1 == "=" and operador2 == "<" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] == condicion1) & (infoOriginal[columna2] < condicion2)
    elif operador1 == "=" and operador2 == ">" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] == condicion1) & (infoOriginal[columna2] > condicion2)
    elif operador1 == "!=" and operador2 == "=" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] != condicion1) & (infoOriginal[columna2] == condicion2)
    elif operador1 == "=" and operador2 == "!=" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] == condicion1) & (infoOriginal[columna2] != condicion2)
    elif operador1 == ">=" and operador2 == ">=" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] >= condicion1) & (infoOriginal[columna2] >= condicion2)
    elif operador1 == ">=" and operador2 == "<=" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] >= condicion1) & (infoOriginal[columna2] <= condicion2)
    elif operador1 == "<=" and operador2 == "<=" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] <= condicion1) & (infoOriginal[columna2] > condicion2)
    elif operador1 == "<=" and operador2 == ">=" and opBool == "and":
        infoFiltrar = (infoOriginal[columna1] <= condicion1) & (infoOriginal[columna2] >= condicion2)
    else:
        infoFiltrar = (infoOriginal[columna1] != condicion1) & (infoOriginal[columna2] != condicion2)

    infoFiltrada = infoOriginal[infoFiltrar]

    if filas == True:
        return len(infoFiltrada)
    else:
        return infoFiltrada
```

Para el primero cálculo, se necesita saber el total de supervivientes, el resultado obtenido, haciendo uso de la función correspondiente, es el que se aprecia en la siguiente figura.

Total de supervivientes

```
#Total de supervivientes
nSuper = totales1Columna(info,"SOBREVIVE","=",True,True)
nSuper
```

```
342
```

En el siguiente cálculo, se necesita saber el total de supervivientes que sean mujeres, el resultado obtenido, es el que se muestra enseguida.

Total de supervivientes mujeres

```
#Total de supervivientes mujeres
nSuperM = totales2Columnas(info,"SOBREVIVE","=",True,"and","SEXO","=","MUJER",True)
nSuperM
```

```
233
```

Para el siguiente cálculo, se requiere conocer el total de

supervivientes que sean hombres, el resultado generado, lo puede observar a continuación.

Total de supervivientes hombres

```
#Total de supervivientes hombres
nSuperH = totales2Columnas(info,"SOBREVIVE","=",True,"and","SEXO","=","HOMBRE",True)
nSuperH
```

```
109
```

En el cálculo que sigue, se trata de conocer el total de supervivientes cuya edad sea mayor o igual a 18 años, , en este caso y en todos los demás cuyo cálculo implique utilizar más de dos columnas del DataFrame, se utiliza la función correspondiente para filtrar por 2 columnas,el resultado que se obtiene se observa enseguida.

Total de supervivientes mayores a 18 años

```
#Total de supervivientes mayores a 18 años
nSuperMay18 = totales2Columnas(info,"SOBREVIVE","=",True,"and","EDAD",">=",18,True)
nSuperMay18
```

```
272
```

En este quinto paso, necesitamos saber el total de supervivientes menores de 18 años, el resultado se muestra en la siguiente imagen.

Total de supervivientes menores a 18 años

```
#Total de supervivientes menores a 18 años
nSuperMen18 = totales2Columnas(info,"SOBREVIVE","=",True,"and","EDAD","<",18,True)
nSuperMen18
```

```
70
```

En este siguiente paso, necesitamos calcular el total de supervivientes cuya edad sea mayor o igual a 50 años, el resultado aparece en la siguiente imagen.

Total de supervivientes mayores a 50 años

```
#Total de supervivientes mayores a 50 años
nSuperMay50 = totales2Columnas(info,"SOBREVIVE","=",True,"and","EDAD",">=",50,True)
nSuperMay50
```

```
31
```

En este cálculo, se requiere conocer el número total de muertos, el resultado aparece a continuación.

Total de muertos

```
#Total de muertos
nMuertos = totales1Columna(info,"SOBREVIVE","=",False,True)
nMuertos
```

```
549
```

Para conocer el resultado de esta paso, es requerido conocer el número total de mujeres muertas, el resultado se muestra en la siguiente figura.

Total de muertos mujeres

```
#Total de muertos mujeres
nMuertosM = totales2Columnas(info,"SOBREVIVE","=",False,"and","SEXO","=","MUJER",True)
nMuertosM
```

```
81
```

En el siguiente cálculo, se necesita conocer el número total de hombres muertos, el resultado se muestra a continuación.

Total de muertos hombres

```
#Total de muertos hombres
nMuertosH = totales2Columnas(info,"SOBREVIVE","=",False,"and","SEXO","=","HOMBRE",True)
nMuertosH
```

```
468
```

En este paso 10, se calculo la tasa se supervivencia de la clase 1, en donde el resultado obtenido es el que se muestra en la siguiente figura.

Tasa de supervivencia de la clase 1

```
#Tasa de supervivencia de la clase 1
nSuperC1 = totales2Columnas(info,"SOBREVIVE","=",True,"and","CLASE","=",1,True)
nC1 = totales1Columna(info,"CLASE","=",1,True)

tasaSuperC1 = (nSuperC1 / nC1) * 100
print("{:.2f}%".format(tasaSuperC1))
```

```
62.96%
```

En el cálculo que sigue, se obtiene la tasa se supervivencia de la clase 2, el resultado del cálculo se muestra a continuación.

Tasa de supervivencia de la clase 2

```
#Tasa de supervivencia de la clase 2
nSuperC2 = totales2Columnas(info,"SOBREVIVE","=",True,"and","CLASE","=",2,True)
nC2 = totales1Columna(info,"CLASE","=",2,True)

tasaSuperC2 = (nSuperC2 / nC2) * 100
print("{:.2f}%".format(tasaSuperC2))
```

```
47.28%
```

Se requiere conocer, en el siguiente paso, la tasa se supervivencia de la clase 3 y para ello se lleva a cabo el cálculo que se muestra en la siguiente imagen.

Tasa de supervivencia de la clase 3

```
#Tasa de supervivencia de la clase 3
nSuperC3 = totales2Columnas(info,"SOBREVIVE","=",True,"and","CLASE","=",3,True)
nC3 = totales1Columna(info,"CLASE","=",3,True)

tasaSuperC3 = (nSuperC3 / nC3) * 100
print("{:.2f}%".format(tasaSuperC3))
```

```
24.24%
```

Para este cálculo, se necesita saber cuál es la tasa de supervivencia de la clase 1 siendo mujer, el resultado es el siguiente.

Tasa de supervivencia de la clase 1 siendo mujer

```
#Tasa de supervivencia de la clase 1 siendo mujer
dfSuperC1 = totales2Columnas(info,"SOBREVIVE","=",True,"and","CLASE","=",1,False)

#El DataFrame dfSuperC1 devuelve los registros filtrados de los supervivientes de la clase 1 y se le pasa a
#la función totales1Columna para filtrar el resultado anterior por la columna SEXO
nVivenC1M = totales1Columna(dfSuperC1,"SEXO","=","MUJER",True)

tasaSuperC1M = (nVivenC1M / nC1) * 100
print("{:.2f}%".format(tasaSuperC1M))
```

```
42.13%
```

En este paso 14, para conocer la tasa de supervivencia de la clase 2 siendo mujer, se lleva a cabo el siguiente cálculo.

Tasa de supervivencia de la clase 2 siendo mujer

```
#Tasa de supervivencia de la clase 2 siendo mujer
dfSuperC2 = totales2Columnas(info,"SOBREVIVE","=",True,"and","CLASE","=",2,False)

#El DataFrame dfSuperC2 devuelve los registros filtrados de los supervivientes de la clase 2 y se le pasa a
#la función totales1Columna para filtrar el resultado anterior por la columna SEXO
nVivenC2M = totales1Columna(dfSuperC2,"SEXO","=","MUJER",True)

tasaSuperC2M = (nVivenC2M / nC2) * 100
print("{:.2f}%".format(tasaSuperC2M))
```

```
38.04%
```

En el siguiente cálculo, se requiere saber cual es la tasa de supervivencia de la clase 3 siendo mujer, el resultado es el que se muestra en laimagen.

Tasa de supervivencia de la clase 3 siendo mujer

```
#Tasa de supervivencia de la clase 3 siendo mujer
dfSuperC3 = totales2Columnas(info,"SOBREVIVE","=",True,"and","CLASE","=",3,False)

#El DataFrame dfSuperC3 devuelve los registros filtrados de los supervivientes de la clase 3 y se le pasa a
#la función totales1Columna para filtrar el resultado anterior por la columna SEXO
nVivenC3M = totales1Columna(dfSuperC3,"SEXO","=","MUJER",True)

tasaSuperC3M = (nVivenC3M / nC3) * 100
print("{:.2f}%".format(tasaSuperC3M))
```

```
14.66%
```

A continuación, inicia el cálculo de probabilidades, para obtener el resultado se hace uso de la regla de Bayes. En primer lugar, se requiere conocer la probabilidad de sobrevivir dado que se es mujer, el resultado al que se llega se muestra en la imagen.

Probabilidad de sobrevivir dado que se es mujer

```
#Probabilidad de sobrevivir dado que se es mujer (usar regla de Bayes)
nMujeres = totales1Columna(info,"SEXO","=","MUJER",True)

#El valor de nSuperM ha sido calculado en "Total de supervivientes mujeres"
pSobreviveM = (nSuperM / nMujeres) * 100
print("{:.2f}%".format(pSobreviveM))
```

```
74.20%
```

La probabilidad de sobrevivir dado que se es hombre, es la siguiente.

Probabilidad de sobrevivir dado que se es hombre

```
#Probabilidad de sobrevivir dado que se es hombre (usar regla de Bayes)
nHombres = totales1Columna(info,"SEXO","=","HOMBRE",True)

#El valor de nSuperH ha sido calculado en "Total de supervivientes hombres"
pSobreviveH = (nSuperH / nHombres) * 100
print("{:.2f}%".format(pSobreviveH))
```

```
18.89%
```

Y por último, el resultado de calcular la probabilidad de sobrevivir dado que se es mujer, y está en la clase 1 se muestra a continuación.

Probabilidad de sobrevivir dado que se es mujer y está en la clase 1

```
#Probabilidad de sobrevivir dado que se es mujer y está en la clase 1 (usar regla de Bayes compuesta)
nClase1M = totales2Columnas(info,"CLASE","=",1,"and","SEXO","=","MUJER",True)

#El valor de nVivenC1M ha sido calculado en "Tasa de supervivencia de la clase 1 siendo mujer"
pSobreviveC1M = (nVivenC1M / nClase1M) * 100
print("{:.2f}%".format(pSobreviveC1M))
```

```
96.81%
```

Conclusiones

Desde el punto de vista de la información cargada en el archivo csv, se puede concluir que la probabilidad de sobrevivir perteneciendo a la clase 1 es de más del 50%, pero si además se es mujer, esa probabilidad aumenta considerablemente (96.81%), en contraste, la tasa de supervivencia al ser mujer y pertenecer a la clase 3 es del 14.66% . Debido a que se llevó a cabo una limpieza de datos y una de estas implicó generar aleatóriamente diferentes edades, la cantidad de personas con respecto a las edades (>= 18, < 18 ó >= 50) puede variar de una corrida de información a otra. Con respecto a las funciones creadas para filtrar la información y obtener cantidades de filas como resultados, estas se pueden mejorar y generar una sola función que lleve a cabo el trabajo de ambas, así como añadir más operadores lógicos a las mismas, ya que por el momento solo se hace uso del operador AND, faltando los operadores OR y NOT.