Eugenia Pérez Pons

PR2- TIPOLOGÍA Y CICLO DE VIDA DE LOS DATOS

LIMPIEZA Y VALIDACIÓN DE LOS DATOS

1. ¿Por qué es importante y qué pregunta/problema pretende responder?

El dataset escogido es el de las muertes y sobrevivientes del Titanic. Me parece un dataset interesante porque es un tema conocido por todos y en el que se puede ver qué probabilidad hubiéramos tenido de sobrevivir nosotros mismos si nos metieramos dentro de uno de los grupos que se van a analizar.

2. Integración y selección de los datos de interés a analizar.

Aunque disponemos de 3 datasets con información (los 3 se encuentran en Github), realmente vamos a cargar solo los de test y train. Cargamos los dos dataset con los diferentes datos que tenemos que tratar. El siguiente paso es imprimir las diferentes columnas para poder ver qué información nos da cada datest. Imprimimos los datos del train y del test para ver que todas las columnas coinciden y que no hay ningún de formato error en las información que hemos cargado. Luego también puede ser interesante en este primer nivel ver un poco la información que nos da cada columna y el tipo de valores con los que nos vamos a encontrar.

```
#importaremos las librerías que necesitamos para tratar los datos
%matplotlib inline

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#EJERCICIO 2
#importamos los datasets y visualizamos la información de las columnas
train = pd.read_csv("train.csv")
test = pd.read_csv("test.csv")

print(train.columns)
print (test.columns)

#vemos los datos de dentro de las columnas
train.sample(4)
```

```
Index([u'PassengerId', u'Survived', u'Pclass', u'Name', u'Sex', u'Age',
       u'SibSp', u'Parch', u'Ticket', u'Fare', u'Cabin', u'Embarked'],
      dtype='object')
Index([u'PassengerId', u'Pclass', u'Name', u'Sex', u'Age', u'SibSp', u'Parch',
       u'Ticket', u'Fare', u'Cabin', u'Embarked'],
      dtype='object')
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
605	606	0	3	Lindell, Mr. Edvard Bengtsson	male	36.0	1	0	349910	15.550	NaN	S
586	587	0	2	Jarvis, Mr. John Denzil	male	47.0	0	0	237565	15.000	NaN	S
883	884	0	2	Banfield, Mr. Frederick James	male	28.0	0	0	C.A./SOTON 34068	10.500	NaN	S
382	383	0	3	Tikkanen, Mr. Juho	male	32.0	0	0	STON/O 2. 3101293	7.925	NaN	S

3. Limpieza de datos

Miramos si lo todos los datos nos valen para entrenar el algoritmo. En el caso de que no nos valgan lo que haremos es limpiarlos (bien eliminando la columna si nos aporta o rellenar con los valores constantes). En el caso en concreto de este dataset vemos que hay varias columnas con datos NaN. Entonces lo que tenemos que hacer cuando empezamos a tratar los datos e incluir el algoritmo es normalizar y unificar los datos porque con valores vacíos no podremos trabajar ni entrenar el algoritmo. Miramos tanto de los datos de test como de los de train.

In [3]:
```
#EJERCICIO 3

#vemos los datos de la tabla
train.describe(include = "all")
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.000000	891	891.000000	204	889
unique	NaN	NaN	NaN	891	2	NaN	NaN	NaN	681	NaN	147	3
top	NaN	NaN	NaN	Graham, Mr. George Edward	male	NaN	NaN	NaN	CA. 2343	NaN	C23 C25 C27	S
freq	NaN	NaN	NaN	1	577	NaN	NaN	NaN	7	NaN	4	644
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.381594	NaN	32.204208	NaN	NaN
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.806057	NaN	49.693429	NaN	NaN
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.000000	NaN	0.000000	NaN	NaN
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.000000	NaN	7.910400	NaN	NaN
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.000000	NaN	14.454200	NaN	NaN
75%	668.500000	1.000000	3.000000	NaN	NaN	38.000000	1.000000	0.000000	NaN	31.000000	NaN	NaN

Miraremos también si hay datos nulos:

```
#como vemos que evarias columnas hay valores NaN lo que hacemos es ver en cuales estan
print(pd.isnull(train).sum())
print(pd.isnull(test).sum())
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

Cuando tenemos impresas las variables de la tabla podemos ver los valores extremos al observar las diferencias entre los valores máximos y mínimos y la mediana. Que será el indicador que nos dirá donde está el promedio y no la media. En nuestros datasets el hecho que haya valores extremos no nos va a afectar porque la información que necesitamos de cada pasajero no nos va a condicionar el hecho de que hubiera algún valor extremo.

4. Análisis de los datos

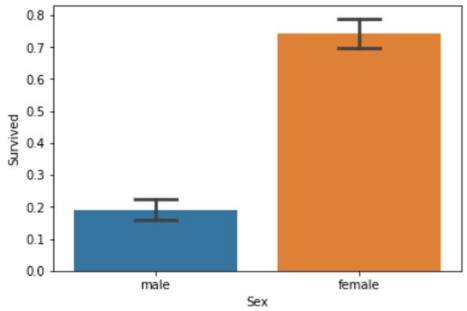
Para poder evaluar y dar respuesta a la pregunta planteada inicialmente que es saber la probabilidad de sobrevivir siendo chica en 28 años en el titanic, los grupos de datos y el tipo de datos que se quieren analizar/comparar son los siguientes valores de los datasets:

- Survived: Int
- Pclass: Int
- Sex: String
- Age: float

En un primer nivel vamos a tratar supervivencia por sexo. Veremos la probabilidad de supervivencia siendo mujer o hombre (habiendo normalizado los valores):

```
: #4-Dibujamos los valores de los que queremos tratar los datos y son
  sns.barplot(x="Sex", y="Survived", data=train, capsize=.2)

  #imprimimos los porcentajes de mujeres y hombres para normalizarlos
  print("Percentage of females who survived:", train["Survived"][train["Sex"] == 'female'].value_counts(normalize = True
  print("Percentage of males who survived:", train["Survived"][train["Sex"] == 'male'].value_counts(normalize = True)[1
```

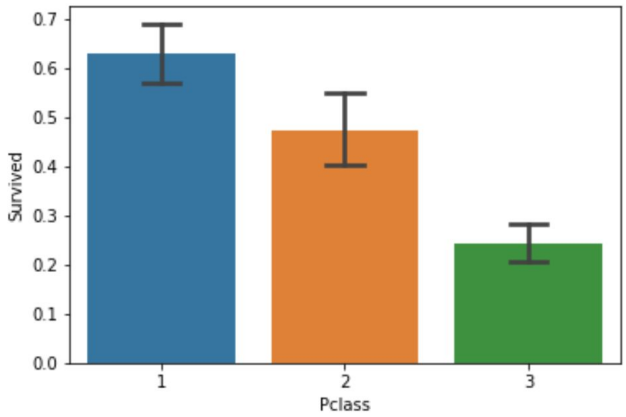
```
('Percentage of females who survived:', 74.20382165605095)
('Percentage of males who survived:', 18.890814558058924)
```



Vemos que tendríamos más probabilidad de sobrevivir siendo mujer. Una vez hecha la primera comparativa haremos luego por por clase. He escogido esta comparativa para ver si la película tenía razón y los pasajeros de primeras clases tenían más probabilidad de sobrevivir.

```
#miramos si tienen más probabilidad de sobrevivir en función de la clase
sns.barplot(x="Pclass", y="Survived", data=train, capsize=.2)

print("Percentage of Pclass = 1 who survived:", train["Survived"][train[
print("Percentage of Pclass = 2 who survived:", train["Survived"][train[
print("Percentage of Pclass = 3 who survived:", train["Survived"][train[
```

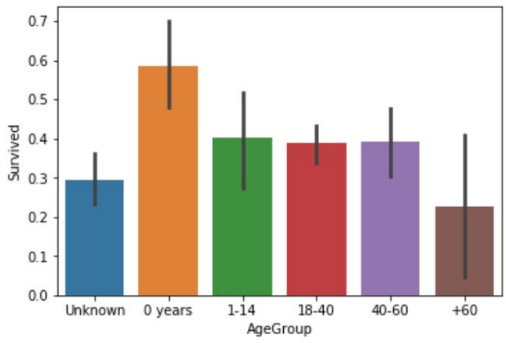
```
('Percentage of Pclass = 1 who survived:', 62.96296296296296)
('Percentage of Pclass = 2 who survived:', 47.28260869565217)
('Percentage of Pclass = 3 who survived:', 24.236252545824847)
```



En el gráfico y las variables podemos observar que los de primera clase tienen más probabilidad de sobrevivir que los de segunda y tercera.

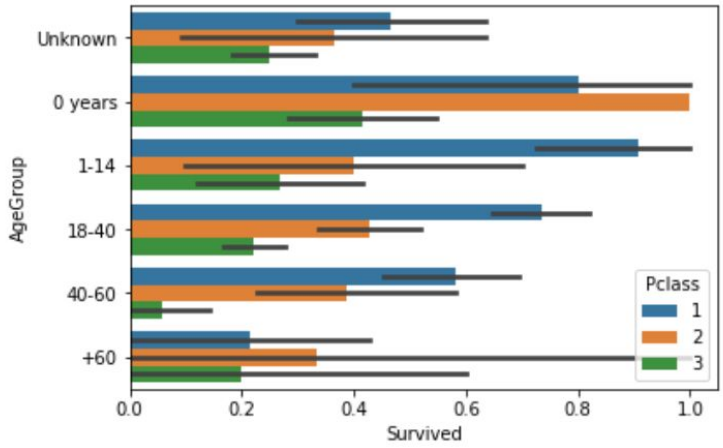
Luego comprobamos la probabilidad de sobrevivir en función de la edad:

```
#miramos la probabilidad de sobrevivir en función de la edad ( manteniendo el test a 40 y el train a 60)

train["Age"] = train["Age"].fillna(-0.6)
test["Age"] = test["Age"].fillna(-0.4)
bins = [-1, 0, 14, 18, 40, 60, np.inf]
labels = ['Unknown', '0 years', '1-14', '18-40', '40-60', '+60']
train['AgeGroup'] = pd.cut(train["Age"], bins, labels = labels)
test['AgeGroup'] = pd.cut(test["Age"], bins, labels = labels)

sns.barplot(x="AgeGroup", y="Survived", data=train)
plt.show()
```



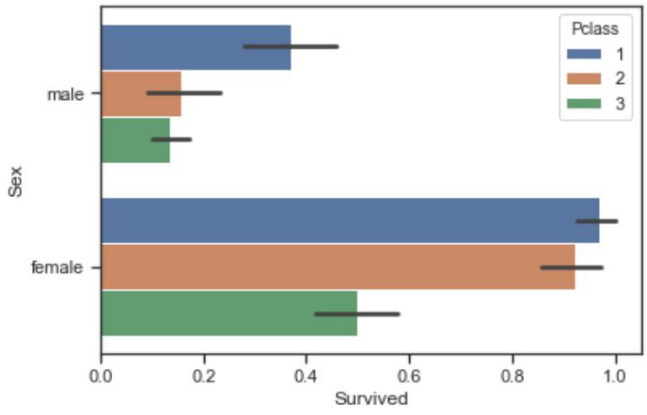
En este caso los niños de 0 años y la gente de entre 1-14 y 40-60 años son los que tenían más probabilidad de sobrevivir.
Luego he comparado también si los de primera clase tenían más probabilidad de sobrevivir que las otras.

```
sns.barplot(x="Survived", y="AgeGroup", hue= "Pclass", data=train)
plt.show()
```



Las mujeres en general son las que sobrevivían más.

```
#probabilidad de sobrevivir en función de la clase y el sexo
sns.barplot(x="Survived", y="Sex", hue= "Pclass", data=train,)
plt.show()
```



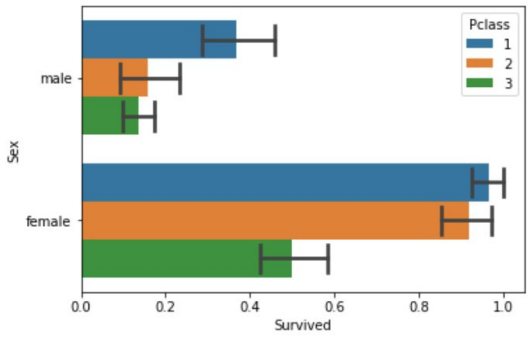
5. **Representación de los resultados a partir de tablas y gráficas.**

   He ido representando con gráficas cada una de las proporciones que he ido haciendo junto con los porcentajes de cada una de las variables.

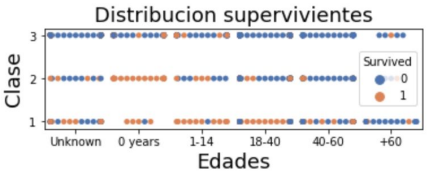
6. **Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?**

   En la última gráfica combinando los datos, los resultados permite ver que realmente hay una relación entre ser mujer, la clase y la edad. Vemos que las personas jóvenes de primera clase son las que tenían más probabilidad de sobrevivir, sin embargo los bebés que iban en segunda clase también iban a sobrevivir.

```
#probabilidad de sobrevivir en función de la clase y el sexo
sns.barplot(x="Survived", y="Sex", hue= "Pclass", data=train, capsize=.2)
plt.show()
```




```
#vemos los supervivientes por edad y clase. 0 cuenta como muertes y 1 es supervivientes

plt.subplot(2,1,2)
sns.swarmplot(x="AgeGroup",y="Pclass",data=train,
              hue="Survived", palette="deep", )

plt.ylabel("Clase", fontsize=18)
plt.xlabel("Edades", fontsize=18)
plt.title("Distribucion supervivientes", fontsize=18)

plt.subplots_adjust(hspace = 0.5, top = 0.9)
plt.show()
```



En el caso que da respuesta a la pregunta, yo siendo mujer ya hemos visto que tenía mucha más probabilidad de sobrevivir que si fuera hombre. Sin embargo, por mi edad lo más óptimo sería que fuera en primera clase porque la probabilidad en segunda clase en mi franja de edad es más baja, y en tercera era prácticamente nula.

7. Código

El código utilizado para este proyecto es Python y están en el siguiente repositorio Github. En el repositorio también se encuentran los CSV utilizados para la práctica.
https://github.com/euperezpons/PR-2

CONTRIBUCIONES	FIRMA
Investigación previa	ME.P
Redacción de las respuestas	ME.P
Desarrollo de código	ME.P