En esta práctica vamos a tener una primera aproximación al aprendizaje automático.

1.- DATASET

Vamos a usar el conjunto de datos de los pasajeros del Titanic, uno de los datasets más utilizados en el aprendizaje automático.

El objetivo es predecir la supervivencia en el desastre del Titanic.

En cualquier supuesto que vayamos a tratar tendremos tres archivos, en nuestro caso concreto estos son:

- titanic_train.csv: contiene información de supervivencia y muerte que usaremos para desarrollar nuestro modelo.
- titanic_test.csv: es el conjunto de pruebas para testear nuestro modelo y que no contiene información de supervivencia y muerte.
- El fichero de resultados, que no desarrollaremos en esta práctica.

En esta práctica nos centraremos en el primero.

2.- ARCHIVOS CSV

Las siglas CSV vienen del inglés "Comma Separated Values" y significan valores separados por comas. Dicho esto, un archivo CSV es cualquier archivo de texto en el cual los caracteres están separados por comas, haciendo una especie de tabla en filas y columnas. Las columnas quedan definidas por cada punto y coma (;), mientras que cada fila se define mediante una línea adicional en el texto

Un archivo CSV suele identificarse con el programa Excel, el cual se basa en cuadrículas que conforman una tabla en filas y columnas. Lo más común es leer archivos CSV desde Excel, ya que el programa (aunque no en las versiones más antiguas) identifica automáticamente los separadores y forma la tabla sin tener que hacer nada por nuestra parte.

Los archivos CSV sirven para manejar una gran cantidad de datos en formato tabla, sin que ello conlleve sobrecoste computacional alguno. Es tremendamente sencillo generar una tabla a partir de un documento de texto, con tan solo delimitar cada celda requerida con un punto y coma (en el caso de Europa) o con una coma (en países de habla inglesa).

3.- GOOGLE COLAB

En esta práctica usaremos Google Colaboratory:

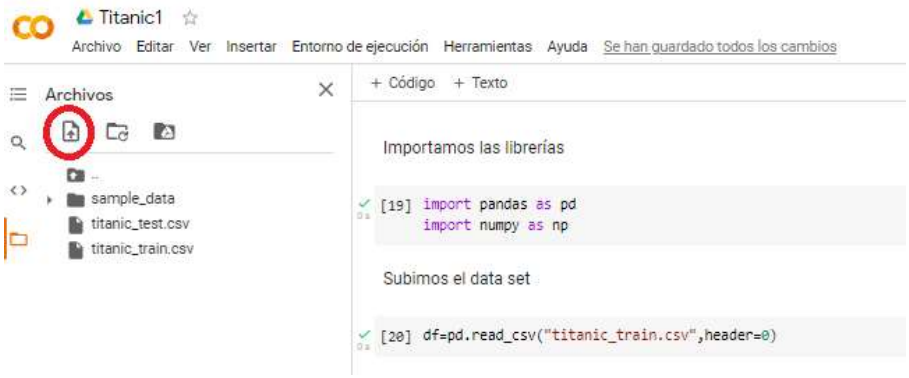
https://colab.research.google.com/notebooks/welcome.ipynb?hl=es

Google Colaboratory es una herramienta que permite ejecutar scripts de Python a través de los servidores de Google. Esta es una de sus ventajes, ya que permite usar los recursos de Google y no solo limitarnos a los recursos de nuestro ordenador.

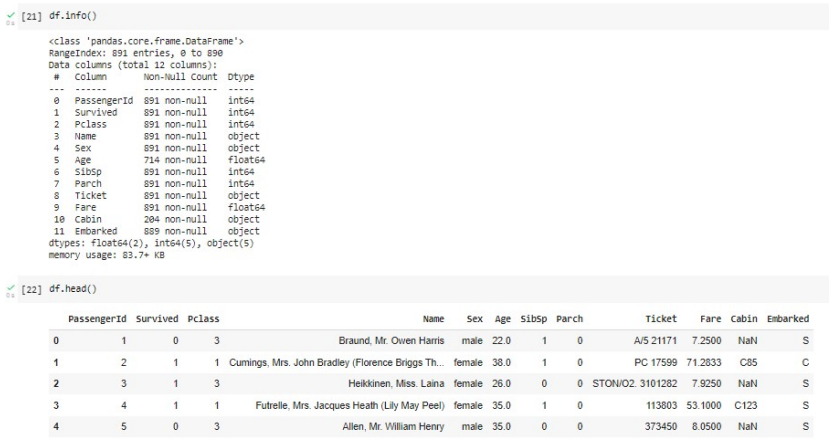
El uso de los servicios de Google Colab es gratuito, aunque la versión gratuita está bastante limitada.

4.-PRÁCTICA

4.1.- Abrir un nuevo cuaderno y subir el dataset:



4.2.- Inspeccionamos el dataset:



```
[21] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
[22] df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

El significado de cada campo aparece en la siguiente tabla:

variable	definición	Valor clave
survival	Supervivencia	0 = No, 1 = Yes
pclass	Clase de entrada	1 = 1st, 2 = 2nd, 3 = 3rd
sex	género	
Age	Edad en años	
sibsp	Número de hermanos / cónyuges a bordo del Titanic Número de hermanos o cónyuges a bordo	
parch	# de padres / hijos a bordo del Titanic Número de padres o hijos a bordo	
ticket	Numero de ticket	
fare	Tarifa de pasajero	
cabin	Número de cabina	
embarked	Puerto de embarque	C = Cherburgo, Q = Queenstown, S = Southampton

4.3.- Estadísticas

Para visualizar qué valores hay en el campo Age y cuantas veces aparece cada dato:

```
df["Age"].value_counts()
```

También vamos a mostrar la media, la mediana y la moda para dicho campo:

```
import statistics as st

print(st.mean(df["Age"].dropna()))
print(st.median(df["Age"].dropna()))
print(st.mode(df["Age"].dropna()))
```

A veces nos interesa saber cómo son de diferentes los valores entre si. Uno de los enfoques más utilizados es coger la media de un conjunto de datos y comparar todos los datos con dicha media. En muchos casos no tiene sentido coger una media y no acompañarla de una medida de dispersión apropiada como la **desviación estándar o desviación media**:

```
np.std(df["Age"].dropna())
```

También se puede usar la **varianza**:

```
np.var(df["Age"].dropna())
```

Con la siguiente sentencia visualizamos cuantos valores no nulos existen en cada columna, su valor medio, la desviación estándar, el valor mínimo y máximo, y los percentiles:

```
df.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

En el caso de la edad, lo que nos indica es que de 0,42 a 20,1250 tenemos el 25% de los datos, hasta 28 el 50% de los datos y hasta 38 el 75% de los datos, lo que nos da idea de lo joven que eran los pasajeros.

4.4.- Representaciones gráficas

4.4.1.- Histogramas

Vamos a dibujar un gráfico que muestre la distribución por edades de los pasajeros del Titanic. Previamente miraremos si el campo "Age" está informado en todos los casos, de no ser así borraríamos los registros de aquellos pasajeros para los que no constase la edad.

Vamos a dibujar un histograma para visualizar la distribución por edades de los pasajeros del Titanic

```
[53] # Mostramos qué valores hay en el campo Age y cuántas veces aparece cada dato
     df["Age"].value_counts()
```

```
24.00    30
22.00    27
18.00    26
19.00    25
30.00    25
         ..
55.50     1
70.50     1
66.00     1
23.50     1
0.42      1
Name: Age, Length: 88, dtype: int64
```

Nos quedaría:

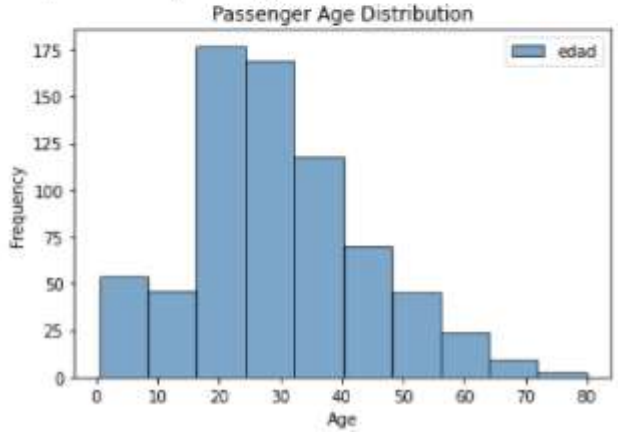
```
plt.title('Passenger Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')

# Eliminar muestras sin datos de edad, sería necesario si hubiese algún campo no informado
df.dropna(subset=['Age'], inplace=True)


plt.hist (df.Age, # datos de dibujo
             bins = 10, # Especifique el intervalo del histograma, dividido en 10 grupos
             color = 'steelblue', # Especificar color de relleno
             edgecolor = 'k', # Especifica el color del borde del histograma
             label = 'edad', # Especificar etiqueta
             alpha = 0.7) # Especificar transparencia

plt.legend()
```

```
<matplotlib.legend.Legend at 0x7fca4997ad10>
```



4.4.2.- subplots

La función matplotlib.pyplot.subplots crea una figura y uno (o varios) conjunto de ejes, devolviendo una referencia a la figura y a los ejes.

Ahora vamos a estudiar su tasa de supervivencia dependiendo del sexo:

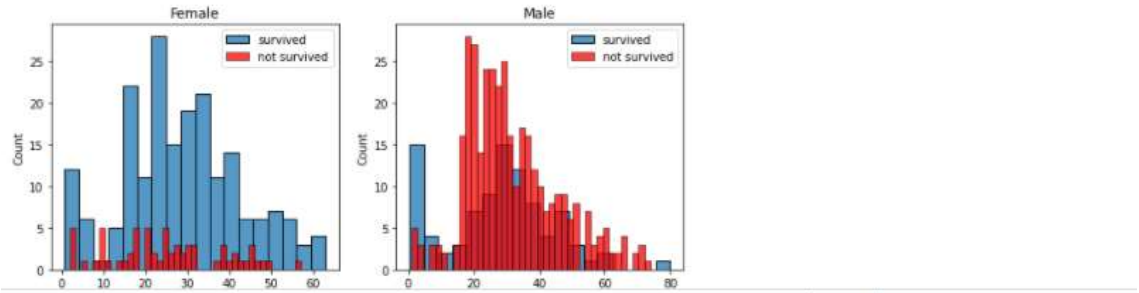
```
import seaborn as sns

survived = 'survived'
not_survived = 'not survived'

fig, axes = plt.subplots(nrows=1, ncols=2,figsize=(10, 4))
women = df[df['Sex']=='female']
men = df[df['Sex']=='male']

ax = sns.histplot(women[women['Survived']==1].Age.dropna(), bins=18, label = survived, ax = axes[0], kde =False)
ax = sns.histplot(women[women['Survived']==0].Age.dropna(), bins=40, label = not_survived, ax = axes[0], kde =False, color='red')
ax.legend()
ax.set_title('Female')

ax = sns.histplot(men[men['Survived']==1].Age.dropna(), bins=18, label = survived, ax = axes[1], kde = False)
ax = sns.histplot(men[men['Survived']==0].Age.dropna(), bins=40, label = not_survived, ax = axes[1], kde = False, color='red')
ax.legend()
ax.set_title('Male')
plt.show()
```



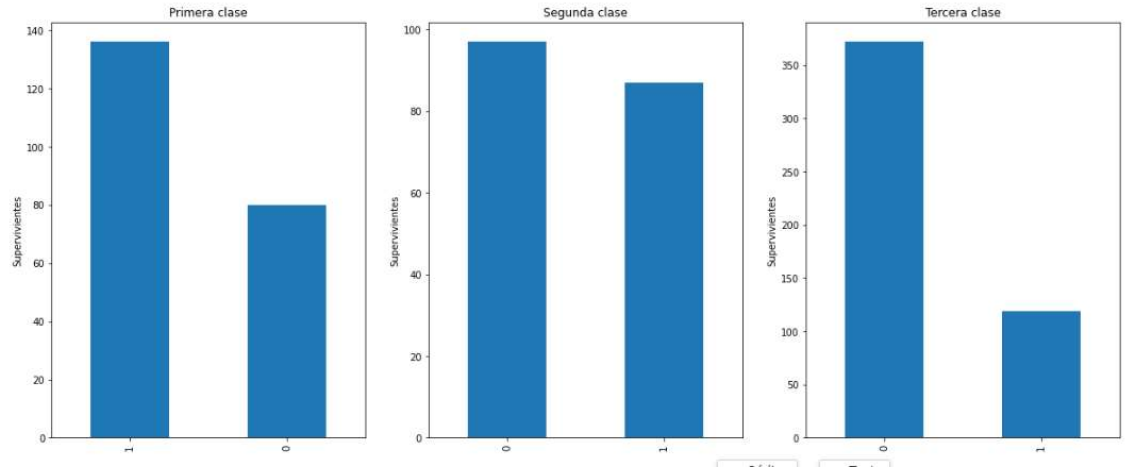
4.4.3.- bar char

Para ver la supervivencia por clase:

```
fig, (ax1, ax2, ax3) = plt.subplots(nrows=1, ncols=3, figsize=(20, 8))

df[df['Pclass']==1].Survived.value_counts().plot(ax=ax1, kind='bar',stacked=False, title='Primera clase', ylabel='Supervivientes')
df[df['Pclass']==2].Survived.value_counts().plot(ax=ax2, kind='bar',stacked=False, title='Segunda clase', ylabel='Supervivientes')
df[df['Pclass']==3].Survived.value_counts().plot(ax=ax3, kind='bar',stacked=False, title='Tercera clase', ylabel='Supervivientes')

plt.show()
```

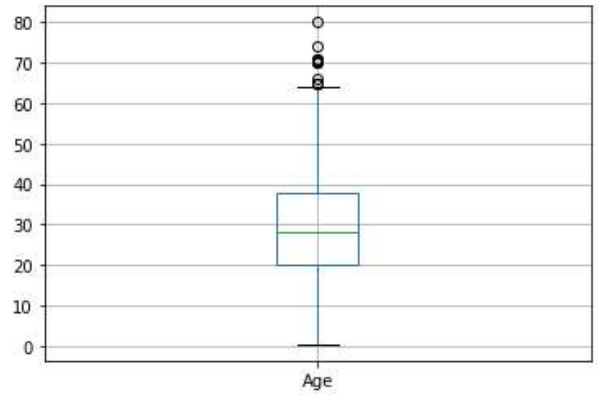


4.4.4.- Diagramas de caja

Un diagrama de caja es una representación de una variable cuantitativa o categórica con el propósito de identificar rápidamente los cuartiles del conjunto de datos.

```
boxplot = df.boxplot(column=['Age'])
boxplot.plot()

plt.show()
```



Las barras de los extremos indican el mínimo (barra inferior) y el máximo (barra superior). Los datos atípicos estarían representados fuera del intervalo del máximo y el mínimo. La parte inferior de la caja es el primer cuartil, la barra del medio de la caja es la mediana o segundo cuartil, la parte superior de la caja es el tercer cuartil.

4.4.5.- Gráfica de dispersión

Los gráficos de dispersión se usan para trazar puntos de datos en un eje vertical y uno horizontal, mediante lo que se trata de mostrar cuánto afecta una variable a otra.

```
scatter_plot=df.plot.scatter(x=['Age'],y=['Pclass'])
scatter_plot.plot()
plt.show()
```