

## **TAREA 5 – PROYECTO ANÁLISIS DE DATOS**

David Esteban Lasso Ordóñez

Universidad Nacional Abierta y a Distancia

Análisis de Datos (202016908\_42)

Breyner Alexander Parra

2024

## TABLA DE CONTENIDO

1	ANÁLISIS EXPLORATORIO.....	3
1.1	Relación entre variables .....	3
1.2	Tendencias .....	4
2	PREPROCESAMIENTO DE DATOS .....	7
2.1	Limpieza de datos.....	7
2.2	Tratamiento de valores faltantes .....	8
2.3	Conversión de columnas categóricas a numéricas .....	8
3	SELECCIÓN DE CARACTERÍSTICAS .....	8
4	DIVISIÓN DEL DATASET EN ENTRENAMIENTO Y PRUEBA .....	10
5	ENTRENAMIENTO DEL MODELO.....	10
6	EVALUACIÓN DEL MODELO .....	11
7	RESULTADOS DEL MODELO .....	11
8	Análisis de los resultados obtenidos .....	13
9	BIBLIOGRAFÍA.....	14

*Anotación: El presente documento únicamente contiene los valores que no son abarcados por la elección de los demás integrantes del grupo.*

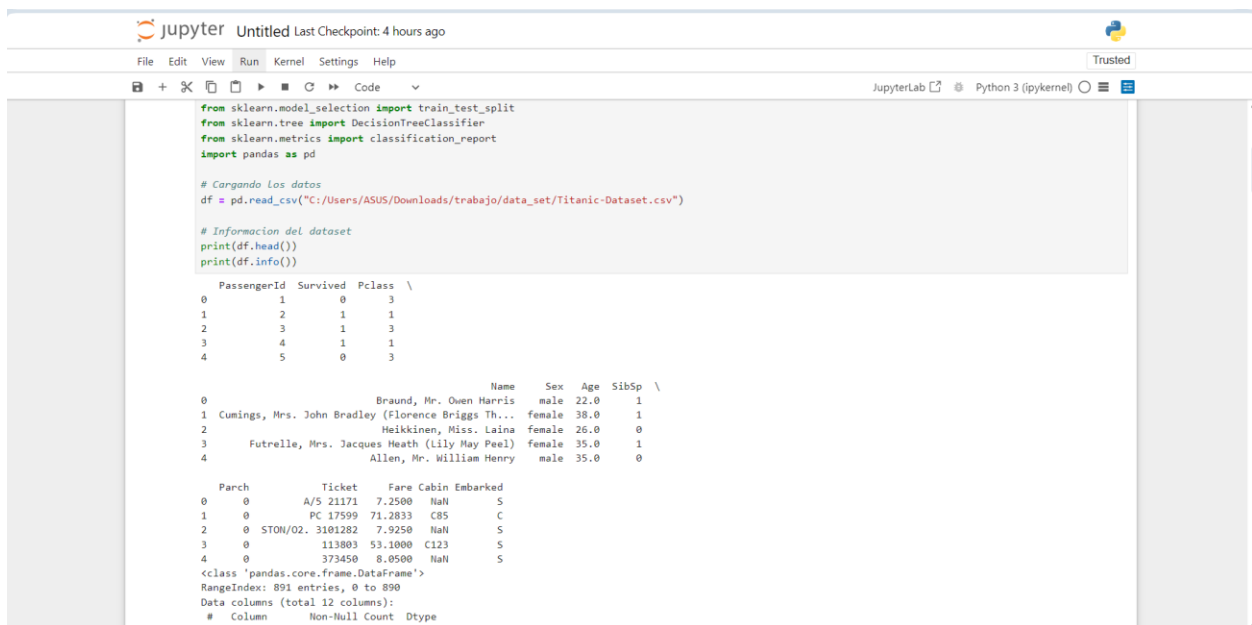
*Gracias por su comprensión.*

# 1 ANÁLISIS EXPLORATORIO

GitHub: [https://github.com/ingDavidLasso/tarea\\_5\\_david\\_lasso\\_G-42.git](https://github.com/ingDavidLasso/tarea_5_david_lasso_G-42.git)

## 1.1 Relación entre variables

El análisis exploratorio de los datos nos ayuda a conocer el contenido del dataset y ver qué patrones o problemas pueden influir en el modelo. Para este proceso cargamos un comando inicial que carga las librerías que necesitamos y también los dataset. Una vez cargados imprime las cuatro primeras filas de cada uno para asegurarnos del correcto funcionamiento de estos, así:



```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
import pandas as pd

# Cargando los datos
df = pd.read_csv("C:/Users/ASUS/Downloads/trabajo/data_set/Titanic-Dataset.csv")

# Información del dataset
print(df.head())
print(df.info())
```

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp
0	1	0	Braund, Mr. Owen Harris	male	22.0	1
1	2	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1
2	3	1	Heikkinen, Miss. Laina	female	26.0	0
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1
4	5	0	Allen, Mr. William Henry	male	35.0	0

Parch	Ticket	Fare	Cabin	Embarked
0	A/5 21171	7.2500	NaN	S
1	PC 17599	71.2833	C85	C
2	STON/O2. 3101282	7.9250	NaN	S
3	113803	53.1000	C123	S
4	373450	8.0500	NaN	S

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
# Column Non-Null Count Dtype
```

**Figura 1.** Carga de datasets e impresión de encabezado por confirmación.

A continuación, exploraremos el contenido e información que contiene el dataset, en consecuencia, distribuiremos el código en la información de estructura mediante el comando `.info()` y el comando `.describe()` para las estadísticas generales del dataset permitiéndonos explorar con mayor precisión la información contenida.

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			
None			

**Figura 2.** Información del dataset.

Análisis estadístico del dataset:					
	PassengerId	Survived	Pclass	Age	SibSp \
count	891.000000	891.000000	891.000000	714.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008
std	257.353842	0.486592	0.836071	14.526497	1.102743
min	1.000000	0.000000	1.000000	0.420000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000
50%	446.000000	0.000000	3.000000	28.000000	0.000000
75%	668.500000	1.000000	3.000000	38.000000	1.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000

	Parch	Fare
count	891.000000	891.000000
mean	0.381594	32.204208
std	0.806057	49.693429
min	0.000000	0.000000
25%	0.000000	7.910400
50%	0.000000	14.454200
75%	0.000000	31.000000
max	6.000000	512.329200

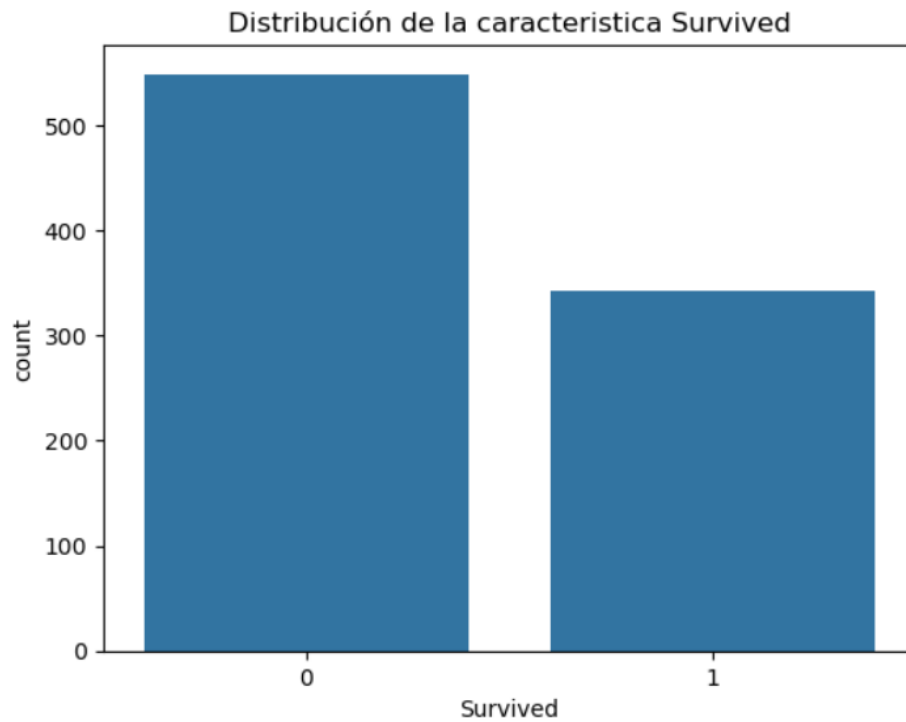
Valores nulos por columna:	
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	177
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	687
Embarked	2
dtype:	int64

**Figura 3.** Estadísticas del dataset.

## 1.2 Tendencias

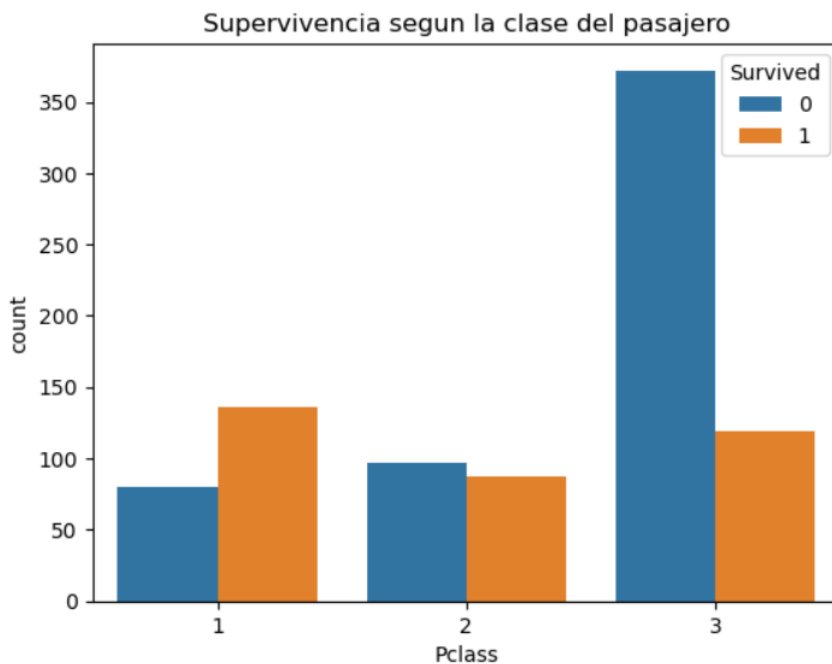
Las tendencias que se identificaron entre los datos del dataset del titanic, e imprimimos la información gráficamente utilizando histogramas para cada tendencia:

- Se evidencio que la característica supervivencia (Survived) tiene más datos en clase 0, lo que quiere decir que más personas no lograron sobrevivir en el agua



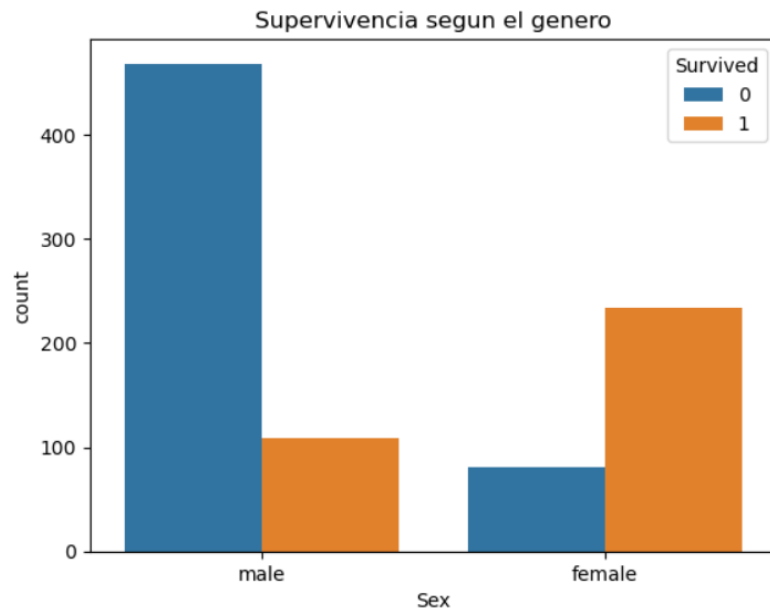
**Figura 4.** Histograma de la característica supervivencia

- Se evidencio que los pasajeros que estaban en la clase 3 mayoritariamente no sobrevivieron.



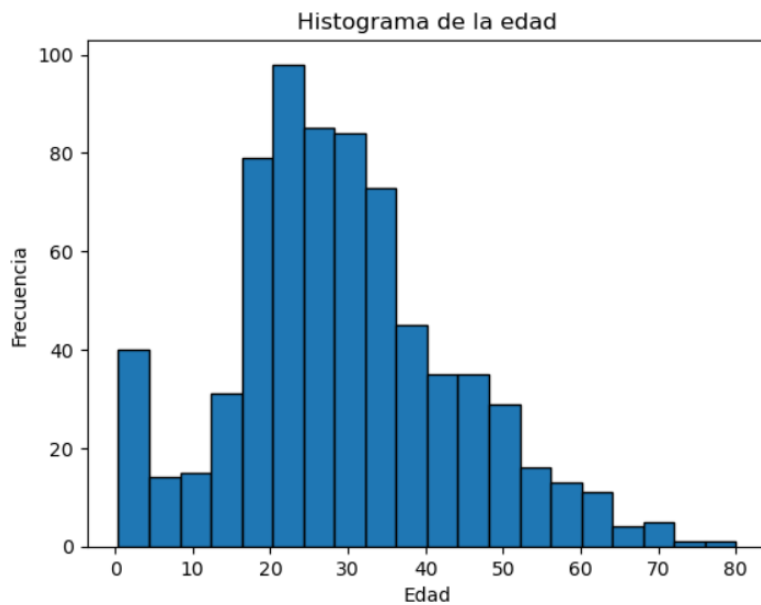
**Figura 5.** Histograma de la supervivencia de acuerdo con la clase del pasajero

- Se evidencio que las personas de género femenino fueron las que sobrevivieron en comparación a las personas de género masculino que mayoritariamente no sobrevivieron.



**Figura 6.** Histograma de la supervivencia según el género.

- En el histograma de distribución de las edades se puede observar que el rango de edad de las personas oscilaba entre 20 y 35 años.



**Figura 7.** Histograma de las edades

## 2 PREPROCESAMIENTO DE DATOS

Esta actividad requiere de múltiples pasos, inicialmente debemos de identificar los valores nulos en nuestro dataset, una vez los hemos identificado y adicionamos en estos espacios un promedio o la moda para que no perjudique nuestra estadística. Debemos de eliminarlos. En este punto ya habríamos realizado la limpieza de los datos, ahora deberemos de tratar los valores faltantes y transformarlos. En consecuencia, el tratamiento y transformación de datos implicara la conversión de variables cualitativas a numéricas como también la disminución de influencia de las variables atípicas por normalización de la información.

### 2.1 Limpieza de datos

Como mencionamos, aquí inicialmente identificamos donde están los valores nulos para cada fila, e imprimimos esta información para que nos sea perceptible. Sin embargo, los eliminamos y reemplazamos para que estadísticamente nuestros resultados sean más fiables respecto a la información verídica.

```
Valores nulos por columna:  
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
```

**Figura 8.** Resultado de investigación de valores nulos

Según el resultado de los valores nulos por columna se evidencia que en las columnas “age, Cabin y embarked” existen valores nulos.

## 2.2 Tratamiento de valores faltantes

Para el tratamiento de datos de las columnas que observamos que tienen datos faltantes para completar los datos faltantes se utilizó la mediana para la columna “age” mediana y para la columna “embarked” se utilizó la moda, eligieron estas técnicas estadísticas para no alterar los datos de cada columna, además se eliminó la columna “Cabin” porque tiene 687 datos nulos, es decir, que la mayoría de sus datos son nulos.

```
[70]: #Se elimina la columna 'Cabin' porque la mayoría de sus datos son nulos (687)
if 'Cabin' in df.columns:
    df = df.drop(columns=['Cabin'])
# En la columna 'Age' se rellenan los valores nulos utilizando la mediana
df['Age'] = df['Age'].fillna(df['Age'].median())

# Para la columna 'Embarked' se rellenan los valores nulos utilizando la moda
df['Embarked'] = df['Embarked'].fillna(df['Embarked'].mode()[0])

# Verificamos si existen valores nulos
print("\nValores nulos después del tratamiento de datos:")
print(df.isnull().sum())

Valores nulos después del tratamiento de datos:
PassengerId    0
Survived        0
Pclass          0
Sex             0
Age             0
SibSp           0
Parch           0
Fare            0
Embarked        0
dtype: int64
```

**Figura 9.** Tratamiento de datos faltantes utilizando la moda y la mediana.

## 2.3 Conversión de columnas categóricas a numéricas

En este punto realizaremos la conversión de variables categóricas a numéricas, como es el caso de las variables “sex, embarked” y para el caso de las variables “Name y ticket” las procedemos a eliminar porque son columnas irrelevantes para el modelo además que estas columnas no son categóricas ni numéricas.

## 3 SELECCIÓN DE CARACTERÍSTICAS

Para observar que la correlación entre las variables para observar cuáles tienen una relación con la variable objetivo que en este caso es survived



```
[68]: #Eliminación de columnas irrelevantes
#Nombre
if 'Name' in df.columns:
    df = df.drop(columns=['Name'])
#Ticket
if 'Ticket' in df.columns:
    df = df.drop(columns=['Ticket'])

#Codificación de Datos categoricos a numericos
#Codificamos la columna sex
var_sex = LabelEncoder()
df['Sex'] = var_sex.fit_transform(df['Sex'])

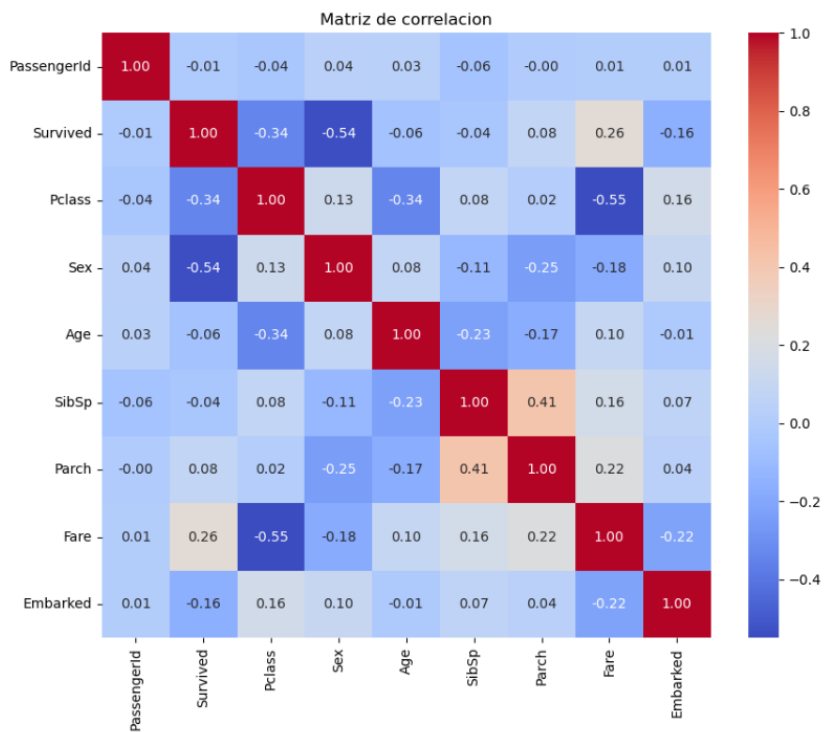
#Codificamos la columna sexEmbarked
var_embarked = LabelEncoder()
df['Embarked'] = var_embarked.fit_transform(df['Embarked'])

print("\nInformación del dataset despues de codificar las columnas categoricas")
print(df.head())
```

```
Información del dataset despues de codificar las columnas categoricas
PassengerId  Survived  Pclass  Sex  Age  SibSp  Parch  Fare  Cabin \
0            1         0       3    1  22.0     1     0   7.2500  NaN
1            2         1       1    0  38.0     1     0  71.2833   C85
2            3         1       3    0  26.0     0     0   7.9250  NaN
3            4         1       1    0  35.0     1     0  53.1000  C123
4            5         0       3    1  35.0     0     0   8.0500  NaN

Embarked
0      2
1      0
2      2
3      2
4      2
```

**Figura 10.** Tratamiento de datos por conversión a variable numérica.



**Figura 11.** Correlación de variables en el dataset.

```
[73]: #Separacion de las caracteristicas
X = df.drop(columns=['Survived'])
y = df['Survived']

print("Variable x:")
print(X.head())

print("\nVariable y:")
print(y.head())

Variable x:
  PassengerId  Survived  Age  SibSp  Parch  Fare  Embarked
0            1         0  22.0     1      0   7.2500         S
1            2         1  38.0     1      0  71.2833         C
2            3         0  26.0     0      0   7.9250         S
3            4         0  35.0     1      0  53.1000         S
4            5         0  35.0     0      0   8.0500         S

Variable y:
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

**Figura 12.** Principales características seleccionadas

## 4 DIVISIÓN DEL DATASET EN ENTRENAMIENTO Y PRUEBA

En este apartado, la división del dataset en entrenamiento y prueba nos permitirá evaluar el rendimiento del modelo en datos que no haya visto antes.

```
[74]: # División del dataset en entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Verificamos la division del dataset
print(f"Entrenamiento (X_train): {X_train.shape}")
print(f"Prueba (X_test): {X_test.shape}")

Entrenamiento (X_train): (712, 8)
Prueba (X_test): (179, 8)
```

**Figura 13.** División del dataset en entrenamiento y test

## 5 ENTRENAMIENTO DEL MODELO

Ahora se realiza el entrenamiento del modelo usando el árbol de decisión y ajustamos sus hiperparámetros, así:

```
[80]: # Creamos el modelo utilizando arboles de decision
model = DecisionTreeClassifier(max_depth=3, random_state=42)

# Entrenamos el modelo
model.fit(X_train, y_train)

print("Modelo ha sido entrenado exitosamente")

Modelo ha sido entrenado exitosamente
```

**Figura 14.** Entrenamiento del modelo.

## 6 EVALUACIÓN DEL MODELO

Para evaluar el modelo en el conjunto de prueba, calcularemos métricas de precision, recall ,accuracy y F1-score.

```
y_pred = model.predict(X_test)
print("Métricas en el conjunto de prueba")
print(classification_report(y_test, y_pred))
```

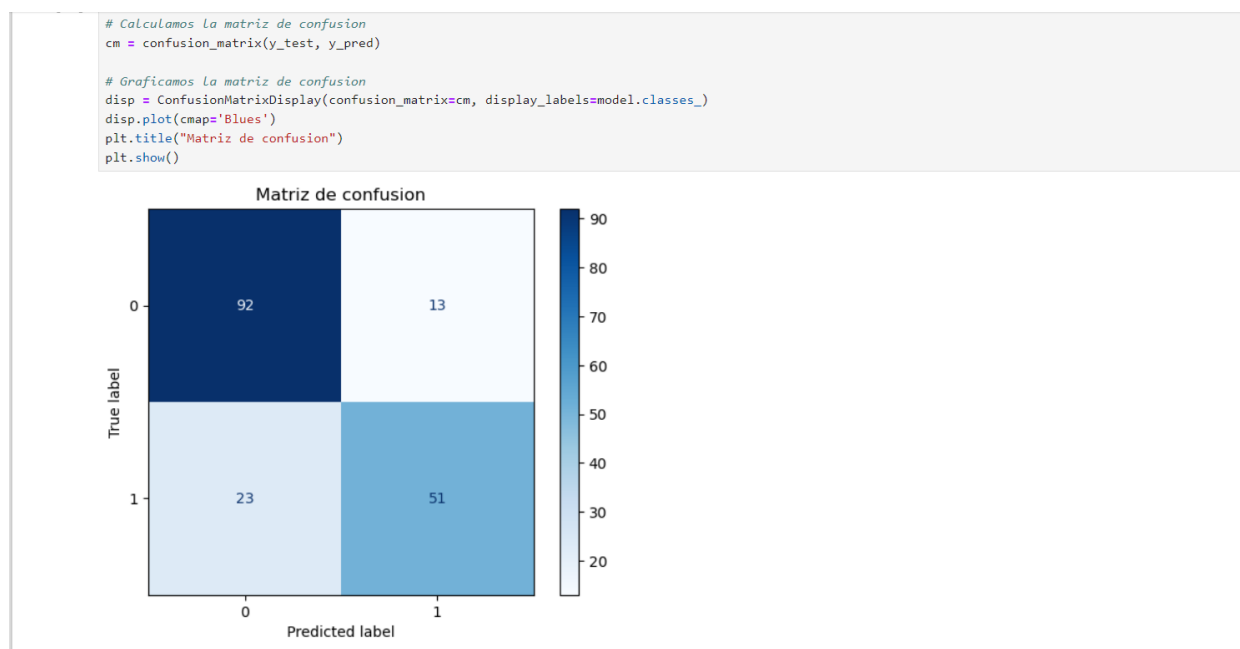
Métricas en el conjunto de prueba				
	precision	recall	f1-score	support
0	0.80	0.88	0.84	105
1	0.80	0.69	0.74	74
accuracy			0.80	179
macro avg	0.80	0.78	0.79	179
weighted avg	0.80	0.80	0.80	179

**Figura 15.** Evaluación del modelo.

## 7 RESULTADOS DEL MODELO

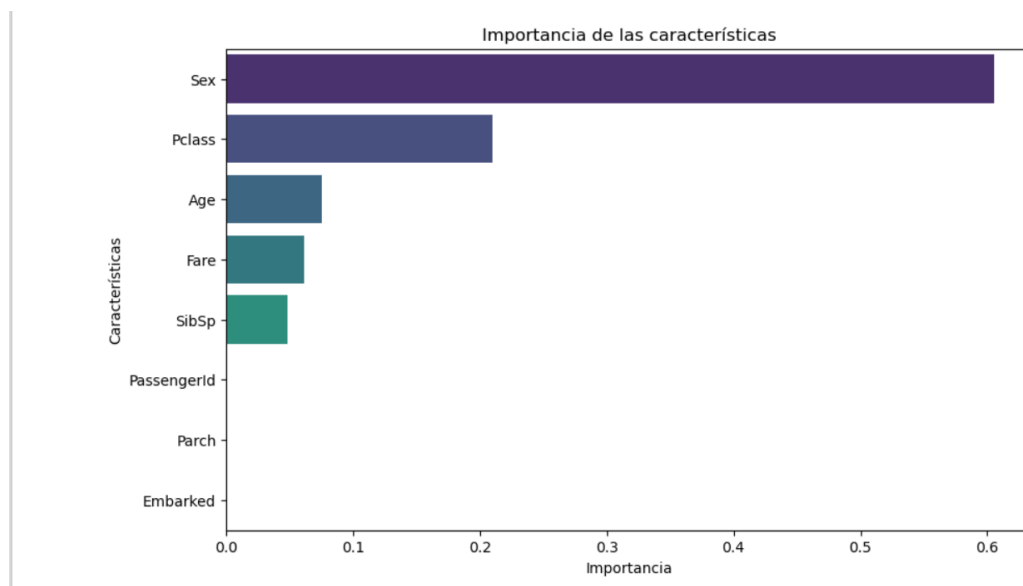
Para observar los resultados obtenidos por el modelo gráficamente podemos observar la matriz de confusión (figura 17) donde el modelo predijo:

- 92 verdaderos positivos
- 51 verdaderos negativos
- 13 falsos positivos
- 23 falsos negativos



**Figura 17.** Matriz de confusión

Podemos observar que la característica más relevante para la supervivencia es la columna “sex” y la columna menos relevante según la supervivencia es “Embarked”.



**Figura 16.** Importancia de las características de acuerdo con la supervivencia

## 8 ANÁLISIS DE RESULTADOS

Según las métricas accuracy, precision, recall y F1-score, podemos saber cómo de bien está funcionando nuestro modelo en términos de predicción de la supervivencia. Que desglosando cada resultado:

- Accuracy (Precisión global): El modelo acertó aproximadamente el 80% de las predicciones en el conjunto de prueba sugiriendo que el modelo es efectivo en la clasificación de la supervivencia de las personas.

Una vez reunida esta información, se determina que el modelo de árbol de decisión tiene un rendimiento sólido en la clasificación de supervivencia de las personas. Además, las métricas indican que es capaz de predecir con precisión al momento de la toma de decisiones.

## 9 BIBLIOGRAFÍA

Carlos Véliz. (2020). Aprendizaje automático. Introducción al aprendizaje profundo. El Fondo Editorial de la Pontificia Universidad Católica del Perú.  
[https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2600876&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp\\_l](https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2600876&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp_l)

David Julian. (2016). Designing Machine Learning Systems with Python. Packt Publishing.  
[https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1218065&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp\\_Cover](https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1218065&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp_Cover)

Giuseppe Bonaccorso. (2018). Machine Learning Algorithms : Popular Algorithms for Data Science and Machine Learning, 2nd Edition: Vol. 2nd ed. Packt Publishing.  
[https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1881497&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp\\_Cover](https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1881497&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp_Cover)

Minguillón, J. Casas, J. y Minguillón, J. (2017). Minería de datos: modelos y algoritmos. Editorial UOC. <https://elibro-net.bibliotecavirtual.unad.edu.co/es/ereader/unad/58656>

Pratap Dangeti. (2017). Statistics for Machine Learning: Build Supervised, Unsupervised, and Reinforcement Learning Models Using Both Python and R. Packt Publishing.  
[https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1560931&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp\\_Cover](https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1560931&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp_Cover)