

Reto Kaggle – Titanic classification

Equipo 2

Arturo Alfaro Gonzalez

Marco Uriel Perez Gutierrez

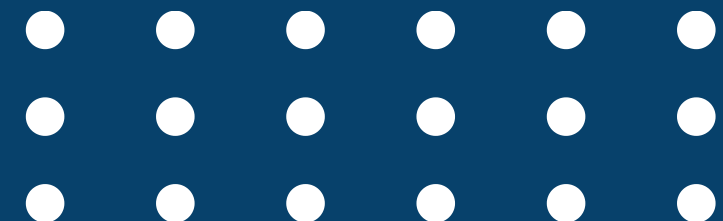
Isaac Jacinto Ruiz

Eli Salomon Martinez Hernandez

Fernando Ortiz Saldaña

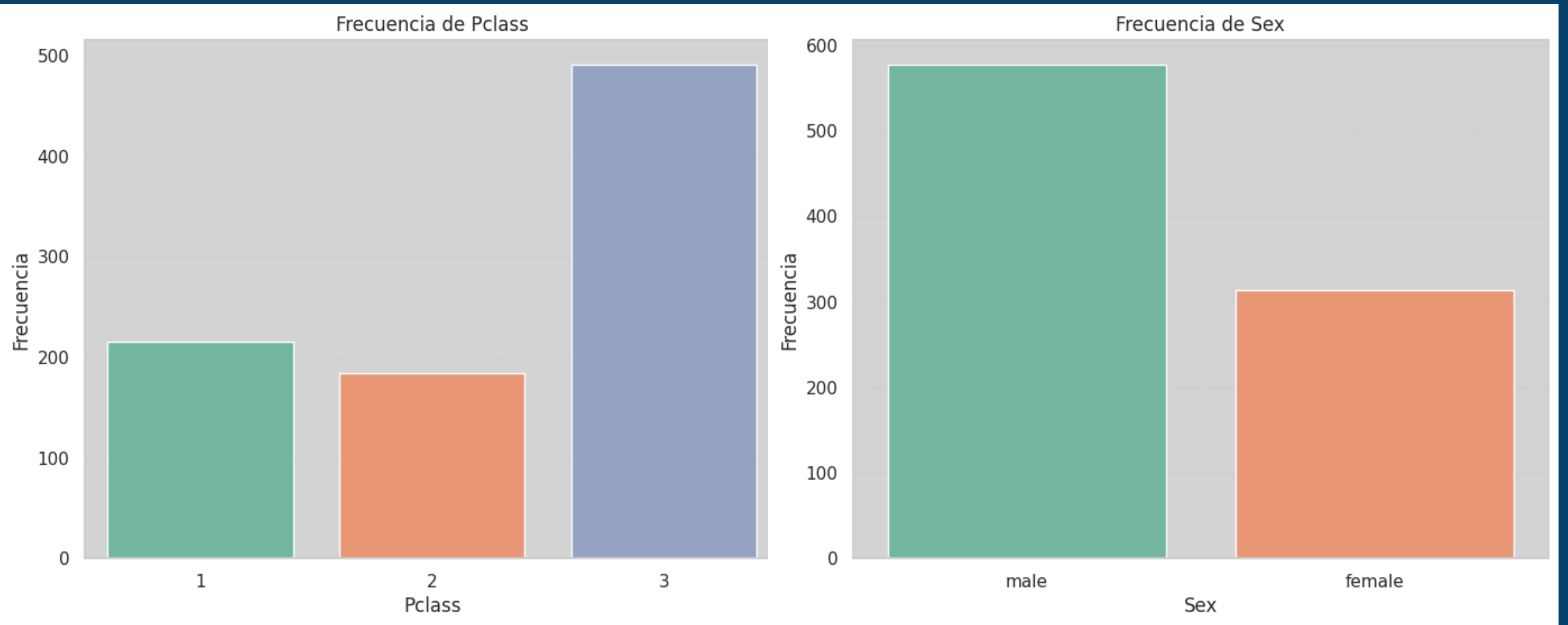
Índice

- 1 Exploración y preprocesamiento de los datos
- 2 Clasificación
- 3 Resultados
- 4 Conclusiones



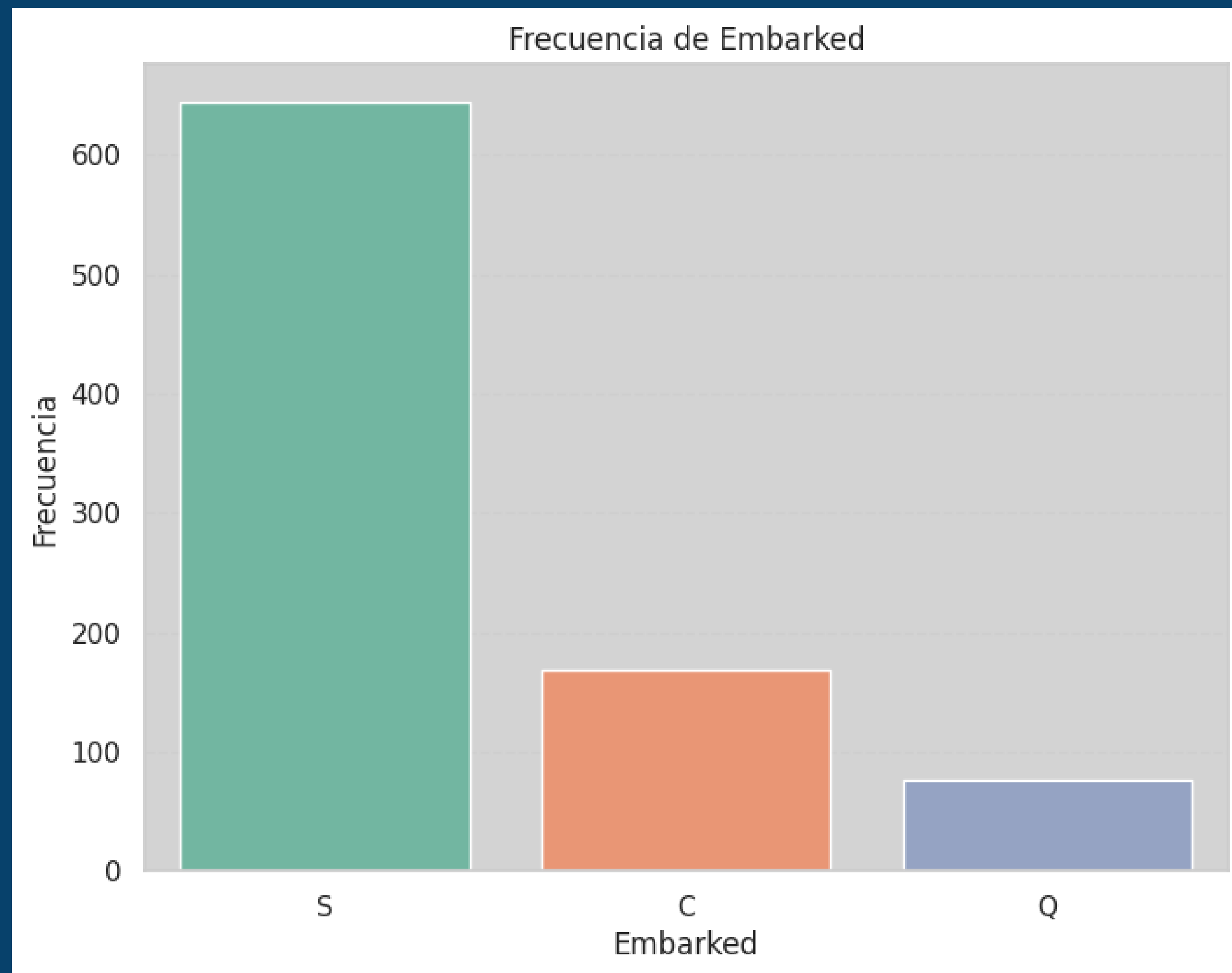
Exploración y preprocesamiento de los datos

Distribución



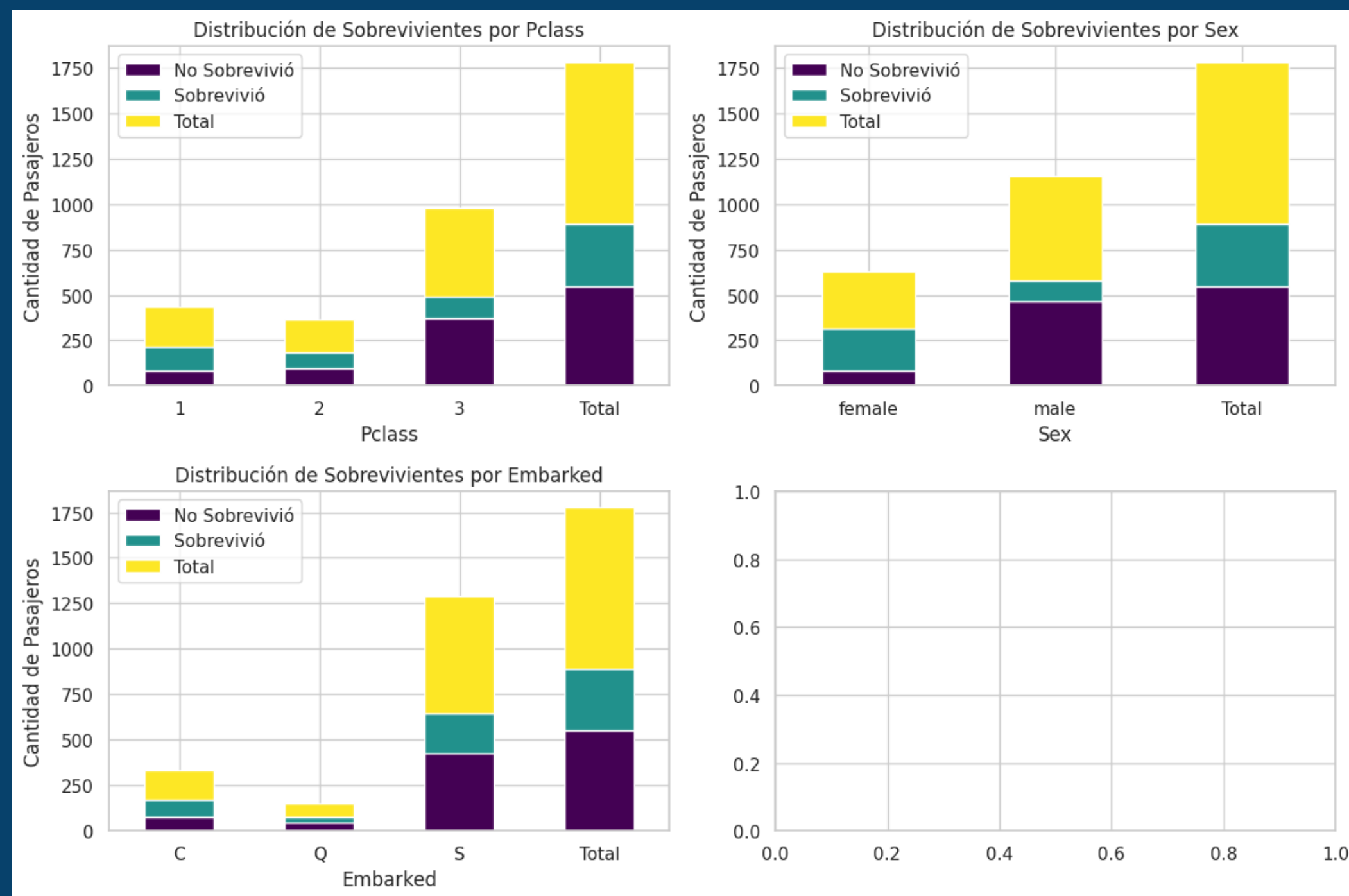
Distribución

En un análisis de los valores categóricos, se encontró que la mayoría estaban ubicados en la clase 3 (70%) y que la mayoría eran hombres (80%). Esto sugiere que la clase y el género pueden ser factores importantes que contribuyen a la supervivencia de los tripulantes.

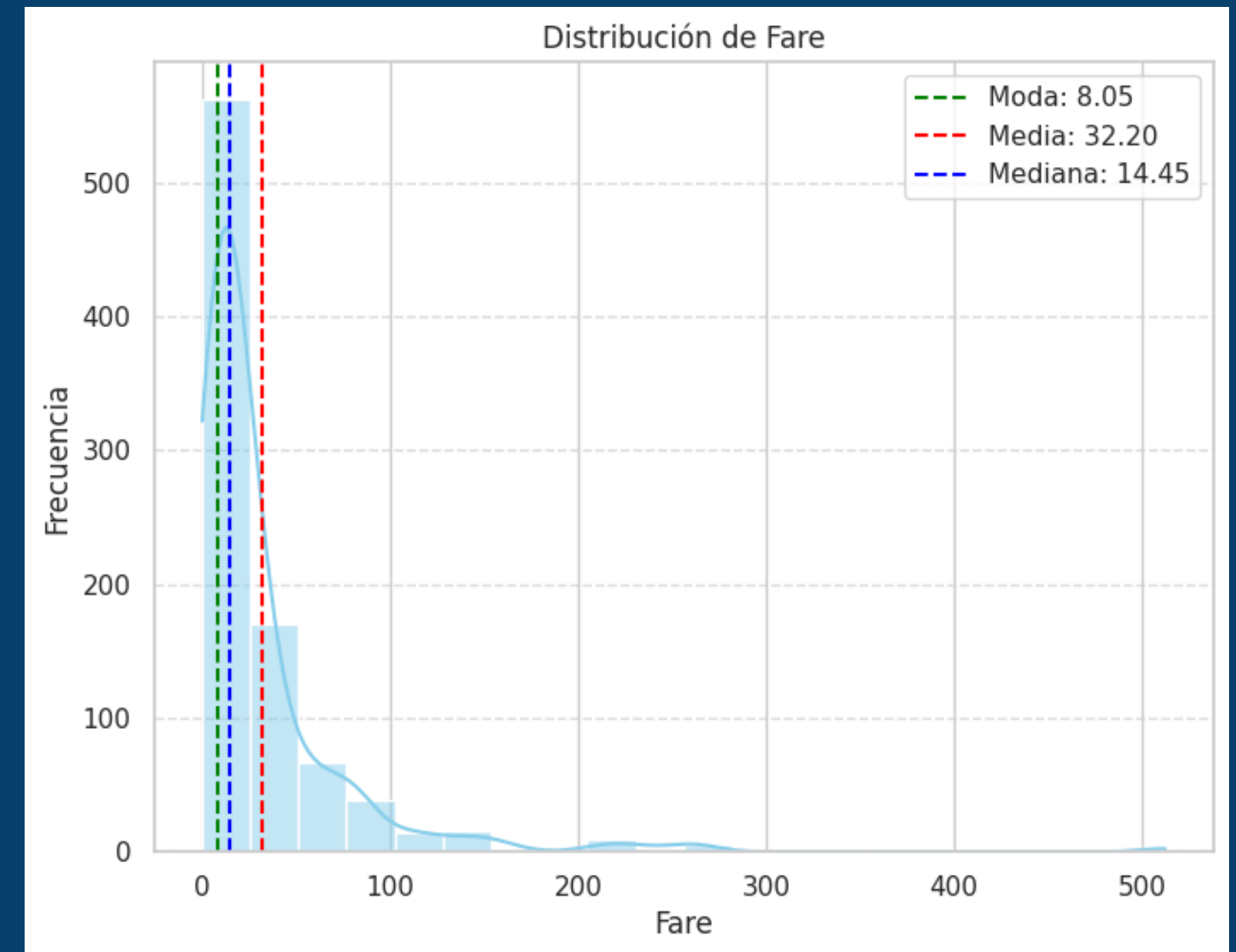
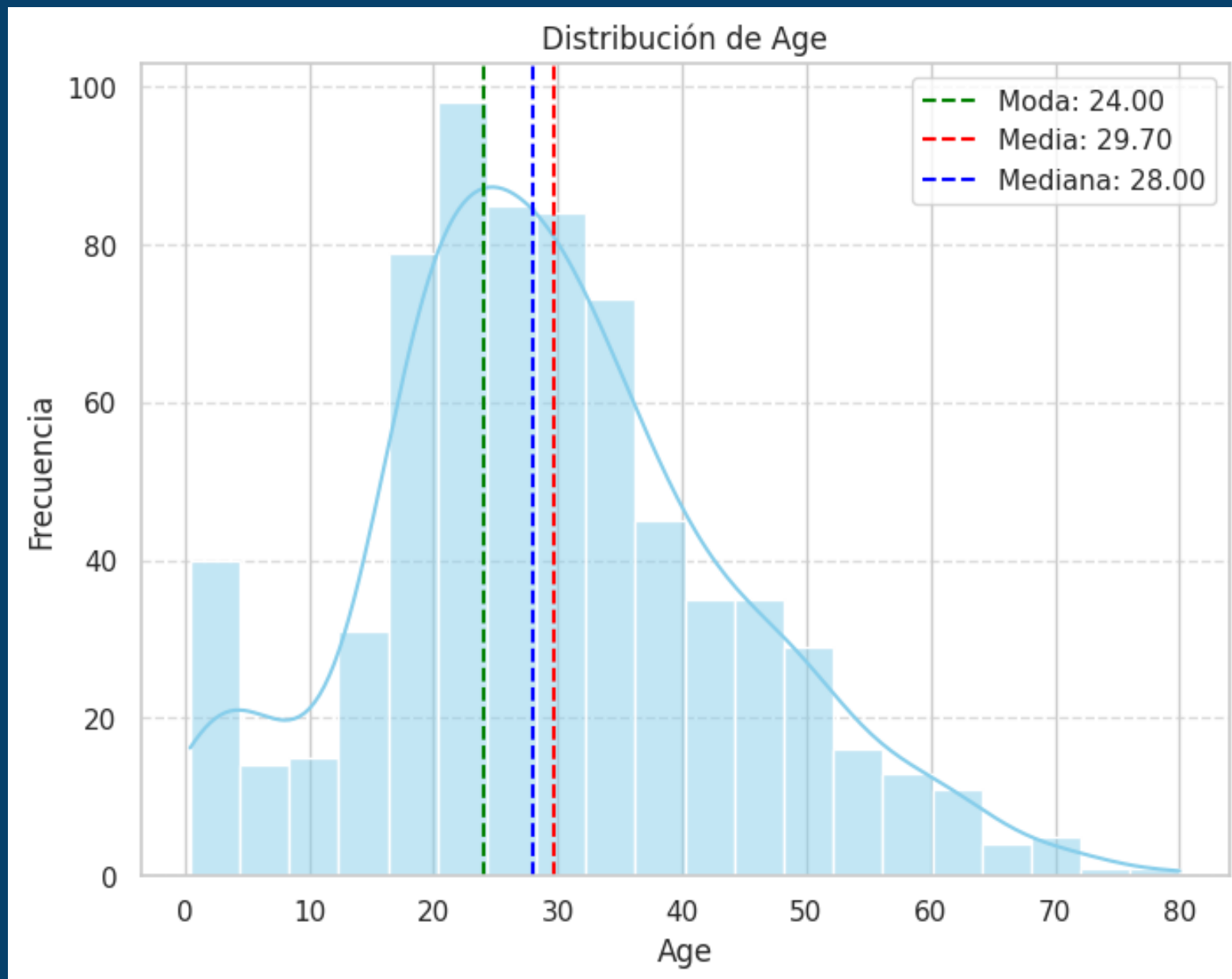


Distribución

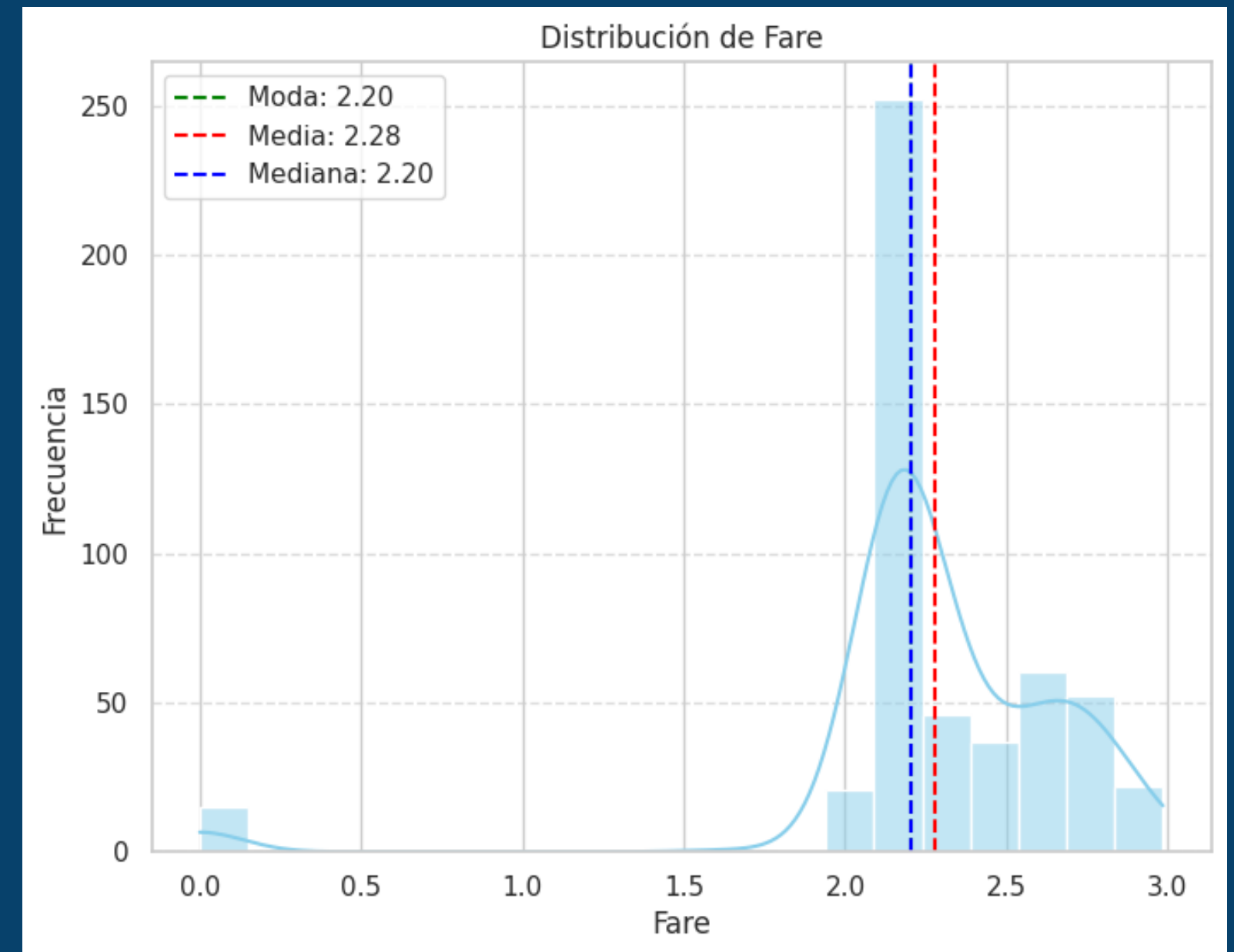
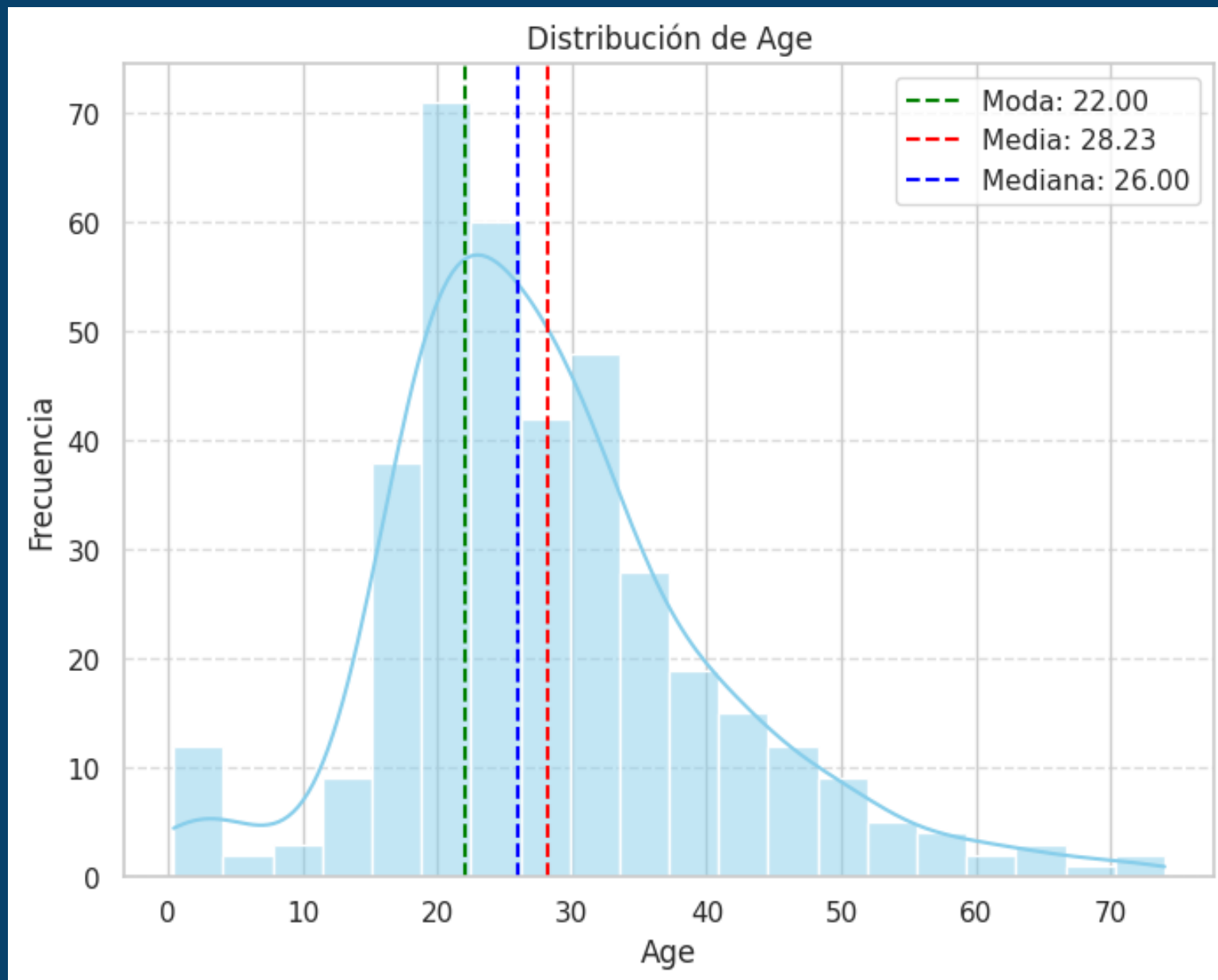
- Los datos no están balanceados puesto que existen más "no supervivientes" que supervivientes.



Distribución

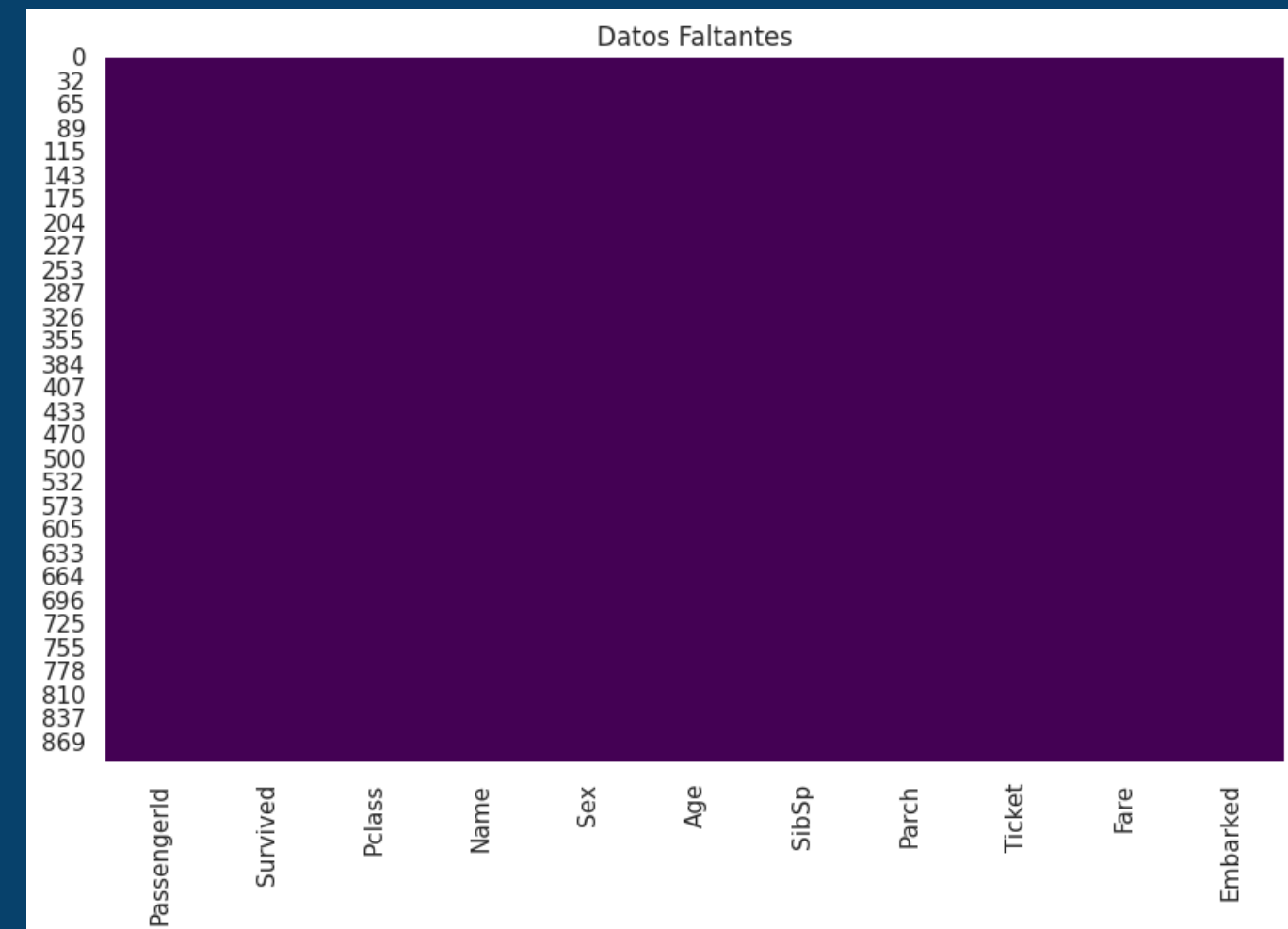
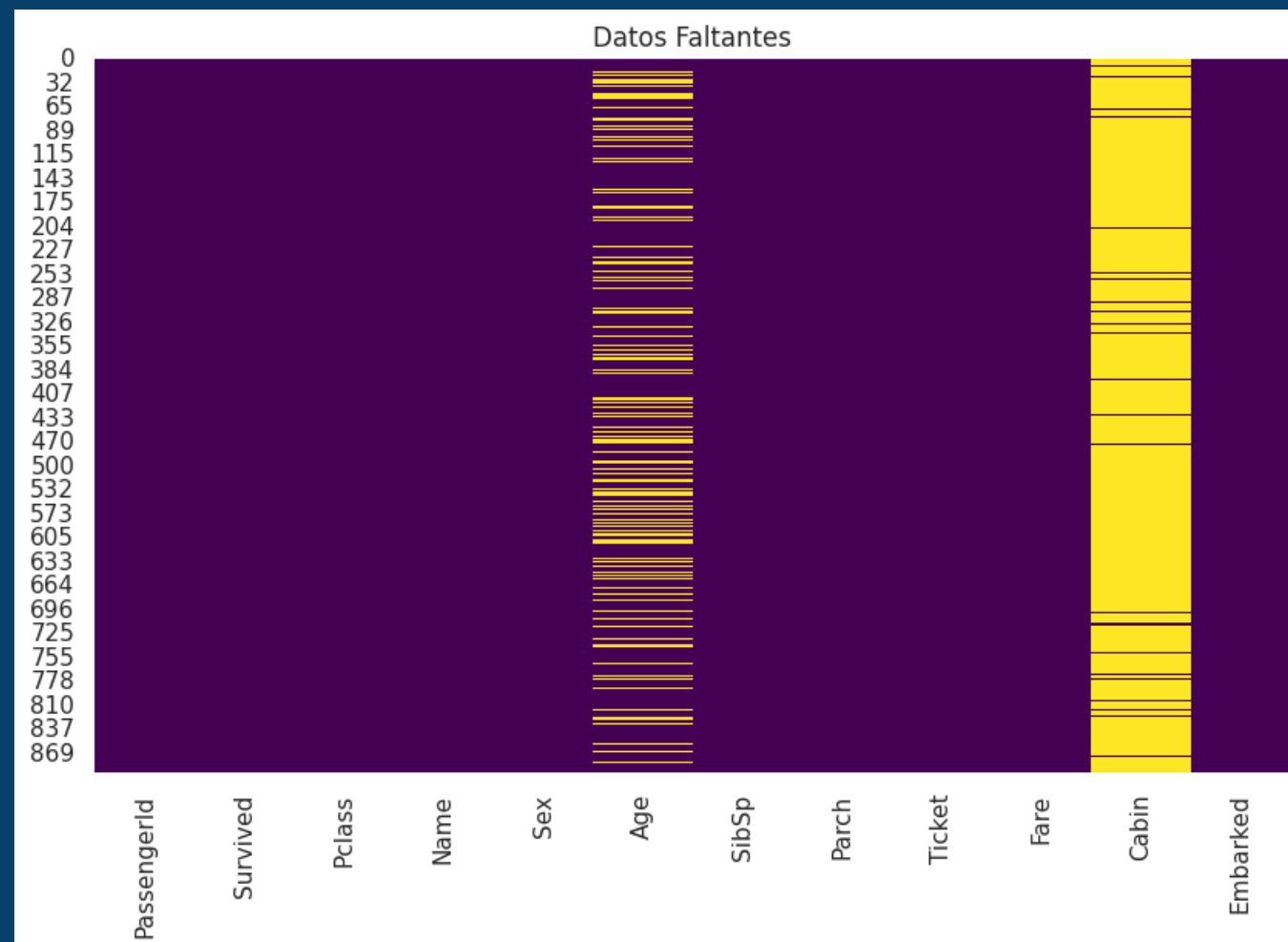


Distribución



Datos Faltantes

- La columna cabin se elimino debido a que tiene muchos datos faltantes
- La columna age se remplazan los valores faltantes con la mediana.
- Los valores faltantes de la columna embarked se rellenaron con la moda.
- La moda es la alternativa más viable para rellenar datos faltantes de variables categóricas.

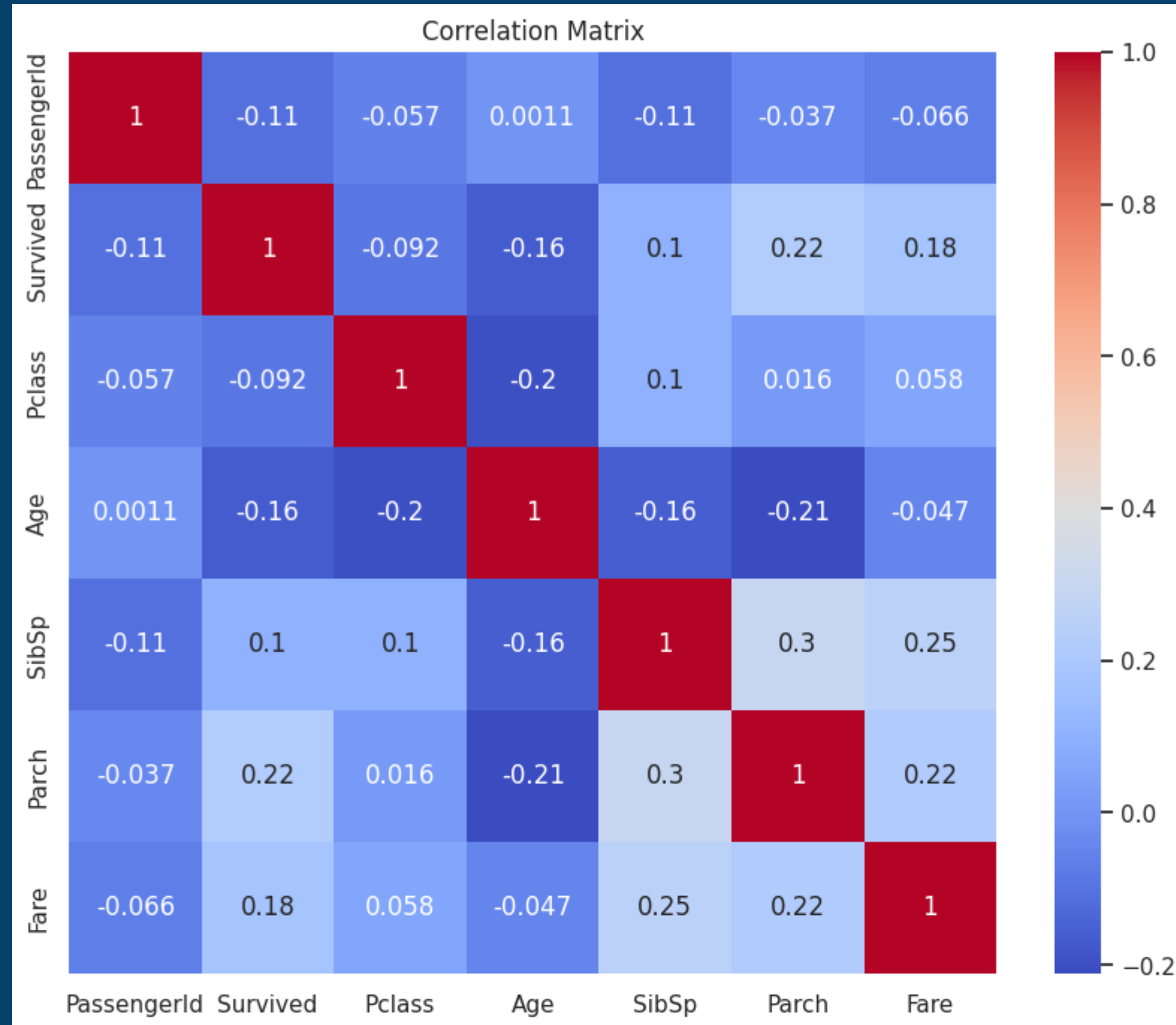


Transformación de datos:

- Los datos categóricos se codificaron para que los modelos de aprendizaje automático pudieran procesarlos.
- Se utilizó label encoding en lugar de one-hot encoding para reducir la dimensionalidad de la tabla.
- Esto mejoró el rendimiento de los modelos al hacerlos más rápidos y fáciles de entrenar.

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	2.110213	2
2	3	1	3	Heikkinen, Miss. Laina	0	26.0	0	0	STON/O2. 3101282	2.188856	2
4	5	0	3	Allen, Mr. William Henry	1	35.0	0	0	373450	2.202765	2
5	6	0	3	Moran, Mr. James	1	26.0	0	0	330877	2.246893	1
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	0	27.0	0	2	347742	2.495954	2
...
882	883	0	3	Dahlberg, Miss. Gerda Ulrika	0	22.0	0	0	7552	2.443798	2
883	884	0	2	Banfield, Mr. Frederick James	1	28.0	0	0	C.A./SOTON 34068	2.442347	2
884	885	0	3	Sutehall, Mr. Henry Jr	1	25.0	0	0	SOTON/OQ 392076	2.085672	2
886	887	0	2	Montvila, Rev. Juozas	1	27.0	0	0	211536	2.639057	2
890	891	0	3	Dooley, Mr. Patrick	1	32.0	0	0	370376	2.169054	1

Análisis de correlación



Metricas

- Accuracy: La exactitud es la fracción de predicciones que son correctas.
- Precision: La precisión es la fracción de predicciones positivas que son realmente positivas.
- Recall: El recall es la fracción de valores positivos reales que son correctamente predichos.
- F1 Score: El F1 es una media ponderada de la precisión y el recall



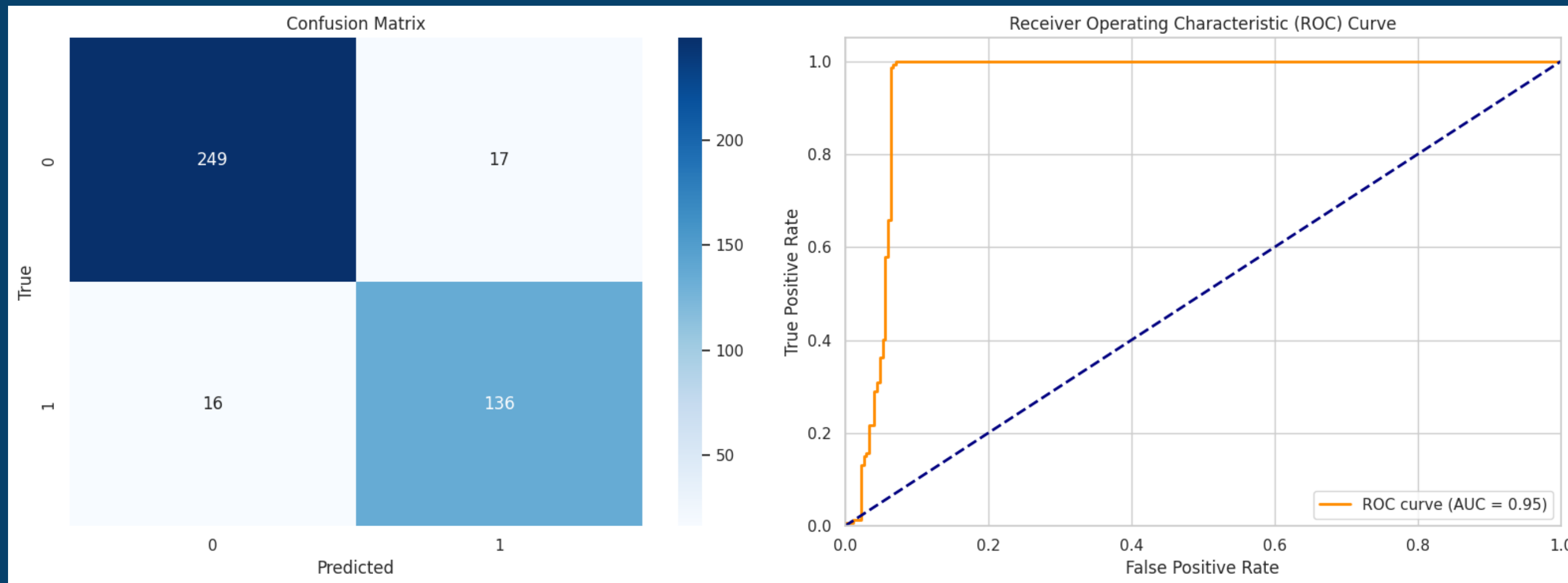
Clasificación

La función primero calcula la matriz de confusión y la curva ROC. La matriz de confusión es una tabla que muestra el número de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos. La curva ROC es una representación gráfica de la sensibilidad (TPR) frente a la especificidad (FPR) para diferentes umbrales.

La función luego grafica la matriz de confusión y la curva ROC. La matriz de confusión se grafica como un mapa de calor, y la curva ROC se grafica como un gráfico de líneas. El AUC (área bajo la curva) de la curva ROC también se calcula y se muestra.

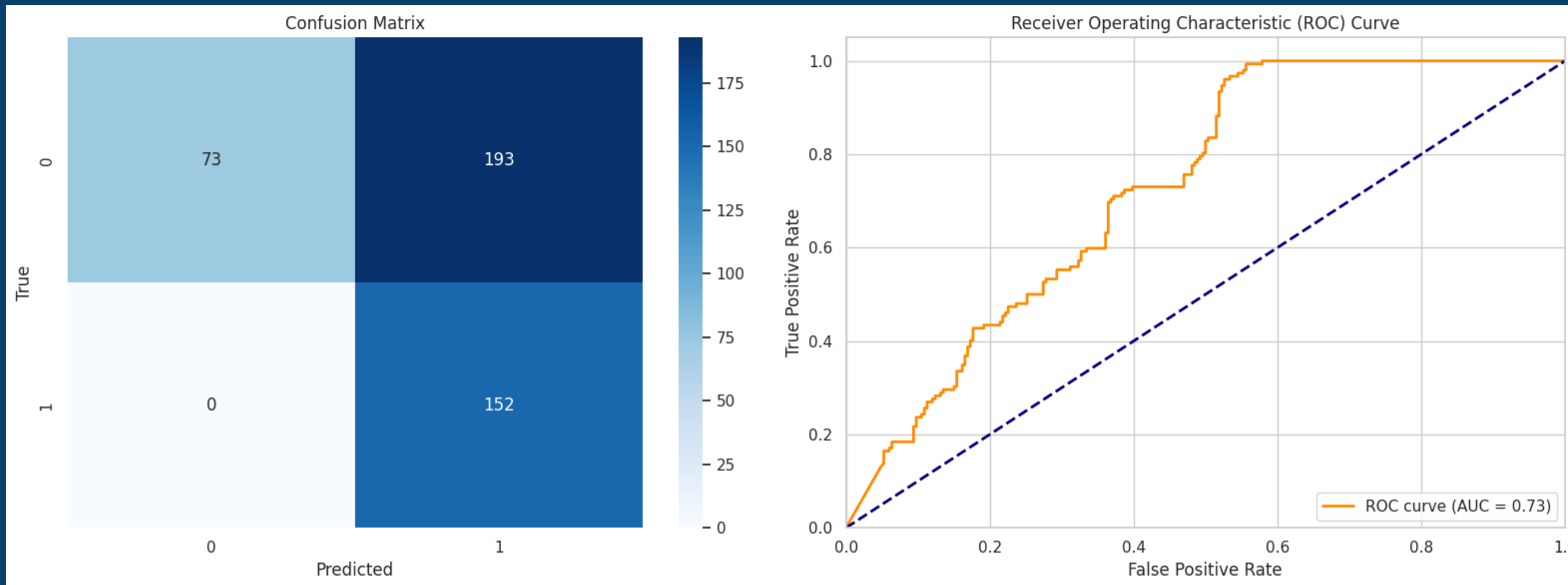
Mientras el AUC sea más grande mejor es el modelo.

- Random Forest: Es un algoritmo robusto y versátil que es adecuado para el desafío de Titanic debido a su resistencia al sobreajuste, su capacidad para manejar datos faltantes y su buen rendimiento inicial.



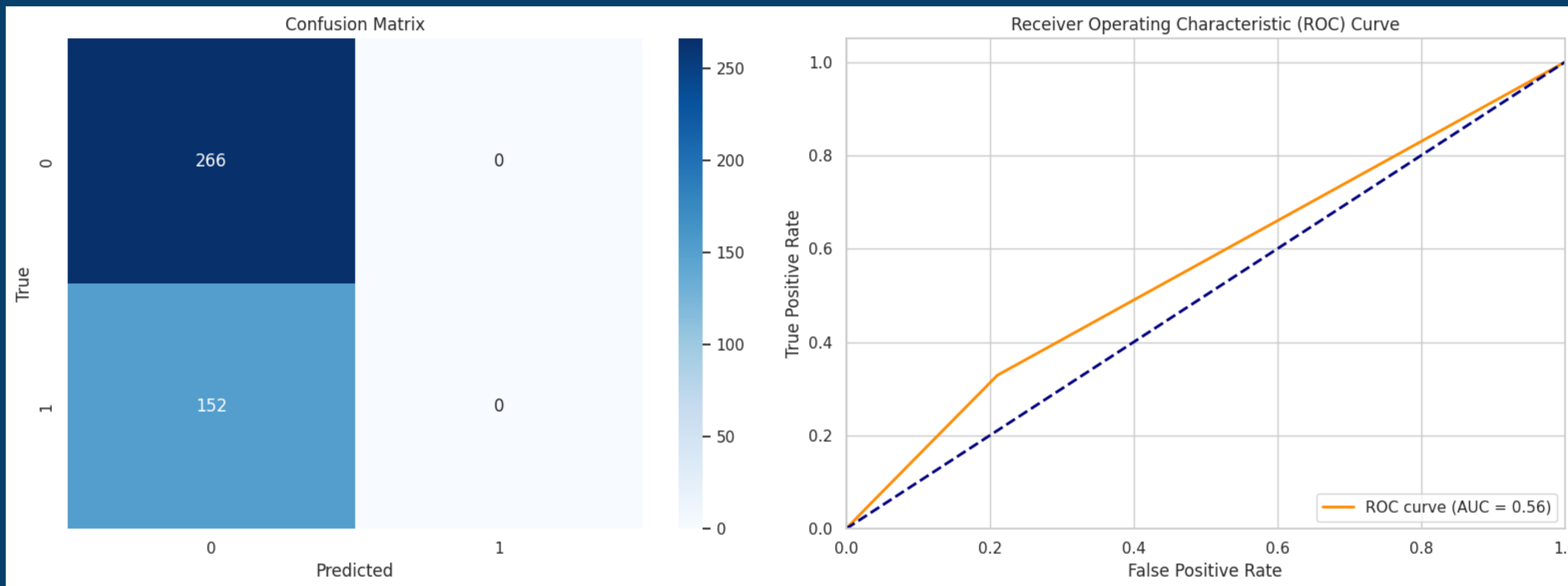
```
Random Forest:  
Accuracy: 0.92  
Precision: 0.92  
Recall: 0.92  
F1-score: 0.92  
=====
```

- Regresión logística: Es un algoritmo adecuado para problemas de clasificación binaria, como el desafío de Titanic. Es eficiente desde el punto de vista computacional, lo que es importante para conjuntos de datos de tamaño moderado a grande.



```
Logistic Regression:  
Accuracy: 0.54  
Precision: 0.80  
Recall: 0.54  
F1-score: 0.50
```

- KNN: Es un algoritmo versátil que puede manejar características numéricas y categóricas. Es simple e interpretable, lo que lo hace adecuado para tareas de clasificación binaria. También permite ajustar el número de vecinos para optimizar el rendimiento.





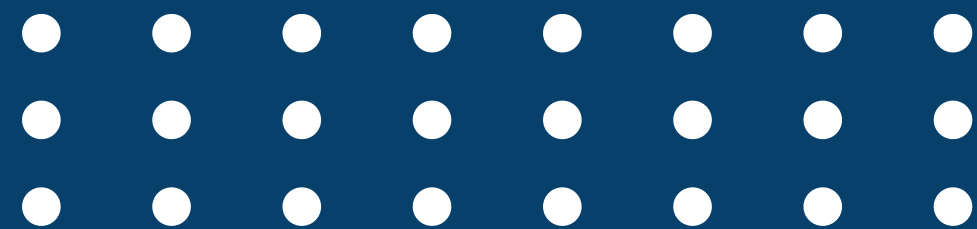
K-Nearest Neighbors:
Accuracy: 0.64
Precision: 0.40
Recall: 0.64
F1-score: 0.49

K-cross validation

```
Valor de k: 2
Puntuación promedio: 0.91
Desviación estándar: 0.00
=====
Valor de k: 5
Puntuación promedio: 0.90
Desviación estándar: 0.04
=====
Valor de k: 7
Puntuación promedio: 0.90
Desviación estándar: 0.02
=====
Valor de k: 10
Puntuación promedio: 0.90
Desviación estándar: 0.04
=====
Valor de k: 12
Puntuación promedio: 0.90
Desviación estándar: 0.04
=====
Valor de k: 15
Puntuación promedio: 0.90
Desviación estándar: 0.05
=====
```

Resultados

Execution and Description		Final Score
	Results.csv Complete · 31m ago	0.75837
	Results.csv Complete · 6h ago	0.69377



Conclusiones

Del proyecto





¡Gracias!