UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE INGENIERÍA

DIPF
POSGRADO
INGENIERÍA

Maestría en Ciencias en
Inteligencia Artificial

K Nearest Neighbors (KNN)

Presenta:
Juan Manuel Aviña Muñoz

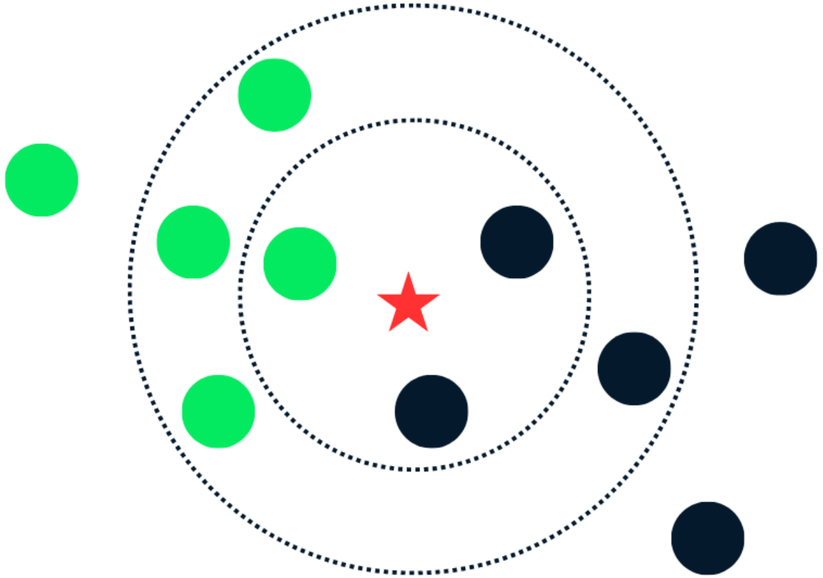
Asesor:
Dr. Marco Aceves Fernández

20 de Octubre del 2023

Índice

Objetivo

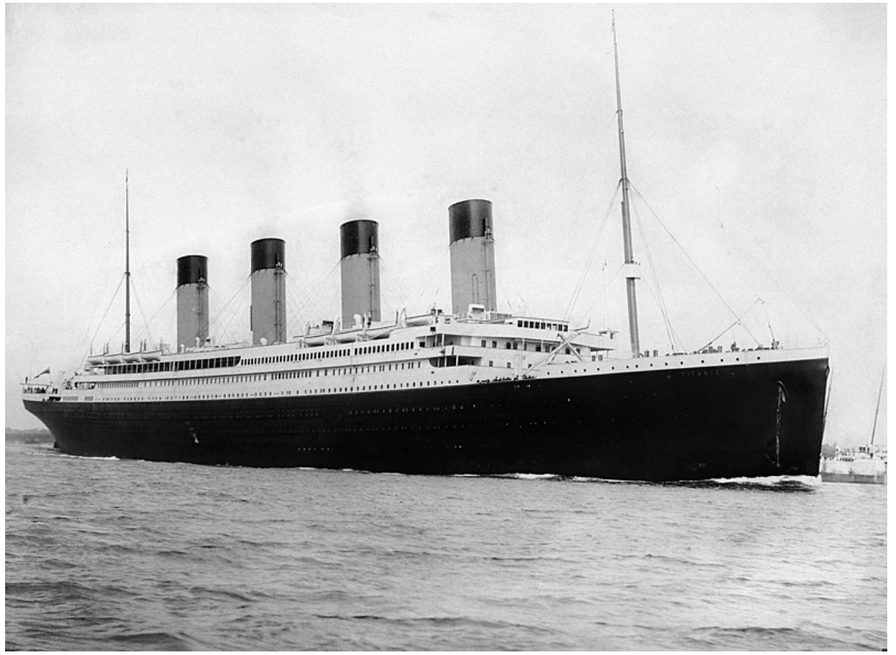
Crear e implementar un clasificador usando KNN



KNN
(DataCamp, 2023)

Justificación

Usar la base de datos del Titanic para implementar un modelo de KNN y generar predicciones de la probabilidad de supervivencia basado en algunas características.



RMS Titanic
(Public domain, 1912).

Fundamentación

KNN es un algoritmo de ML muy utilizado debido a su simpleza para tareas de regresión y clasificación.

Sin embargo, tiene algunas limitaciones.

Fundamentación

Pasos:

1. Inicialización
2. Selección de “K”
3. Calcular distancias
4. Seleccionar vecinos
5. Votación
6. Regresión
7. Predicción
8. Repetir 3-7

Justificación

Ventajas:

- Fácil de entender e implementar.
- La fase de entrenamiento es rápida.
- Regresión y Clasificación.
- No toma en cuenta la distribución de los datos.

Desventajas:

- El algoritmo puede ser lento.
- Sensible a escala de características.
- Puede ser complicado seleccionar una K apropiada.

Métodos


```
Total instances: 1309
Total features: 12
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.00	1	0	A/5 21171	7.25	S	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.00	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.00	0	0	STON/O2. 3101282	7.925	S	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.00	1	0	113803	53.1	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.00	0	0	373450	8.05	S	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...
1304	1305	0	3	Spector, Mr. Woolf	male	0.00	0	A.5. 3236		8.05	S	NaN
1305	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.00	0	0	PC 17758	108.9	C105	C
1306	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.50	0	0	SOTON/O.Q. 3101262	7.25	S	NaN
1307	1308	0	3	Ware, Mr. Frederick	male	0.00	0	359309		8.05	S	NaN
1308	1309	0	3	Peter, Master. Michael J	male	1.00	1	2668		22.3583	C	NaN

1309 rows × 12 columns

Dataset data display
(Creación propia, 2023)

Métodos


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

from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import ConfusionMatrixDisplay, confusion_matrix, accuracy_score, recall_score
from sklearn.metrics import precision_score, f1_score

from sklearn.model_selection import train_test_split

from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2

from utils import dimensionality_reduction

pd.options.display.float_format = '{:.2f}'.format

pie_colors = ['#F08080', '#CCCCFF', '#9FE2BF']
countplot_colors = ['#a6b1f7', '#ffcc99', '#DAF7A6']
violinplot_colors = ['#FF5733', '#dbff33', '#ffbd33', '#33dbff']
```

Librerías
(Creación propia, 2023)

Resultados

Inspección de la base de datos.

```
PassengerId       0
Survived          0
Pclass            0
Name              0
Sex               0
Age               0
SibSp             0
Parch             0
Ticket            0
Fare              0
Cabin           241
Embarked       1039
dtype: int64
```

Datos faltantes
(Creación propia, 2023).

```
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  1309 non-null   int64
 1   Survived     1309 non-null   int64
 2   Pclass       1309 non-null   int64
 3   Name         1309 non-null   object
 4   Sex          1309 non-null   object
 5   Age          1309 non-null   float64
 6   SibSp        1309 non-null   int64
 7   Parch        1309 non-null   object
 8   Ticket       1309 non-null   object
 9   Fare         1309 non-null   object
 10  Cabin        1068 non-null   object
 11  Embarked     270 non-null    object
dtypes: float64(1), int64(4), object(7)
memory usage: 122.8+ KB
```

Datos faltantes
(Creación propia, 2023).

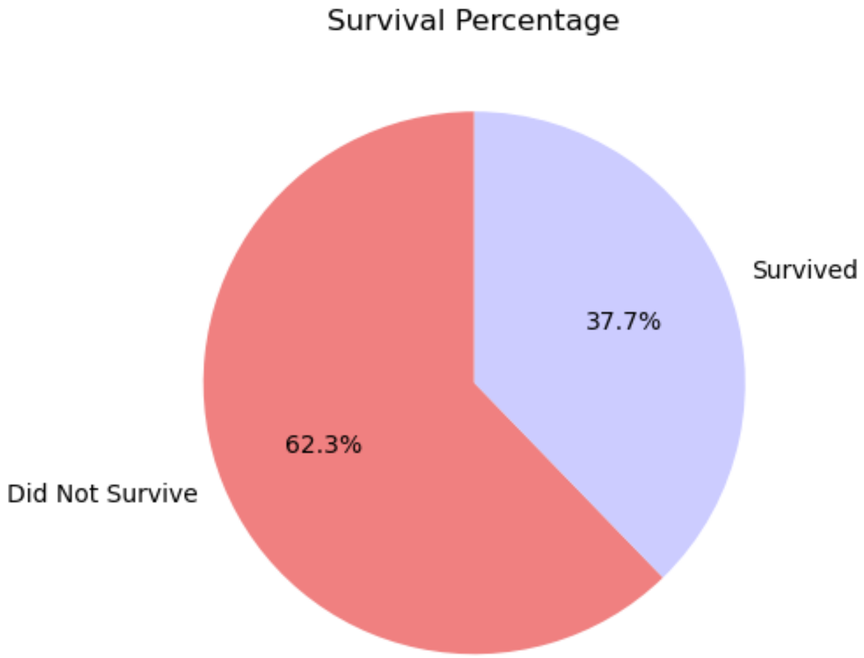
Resultados

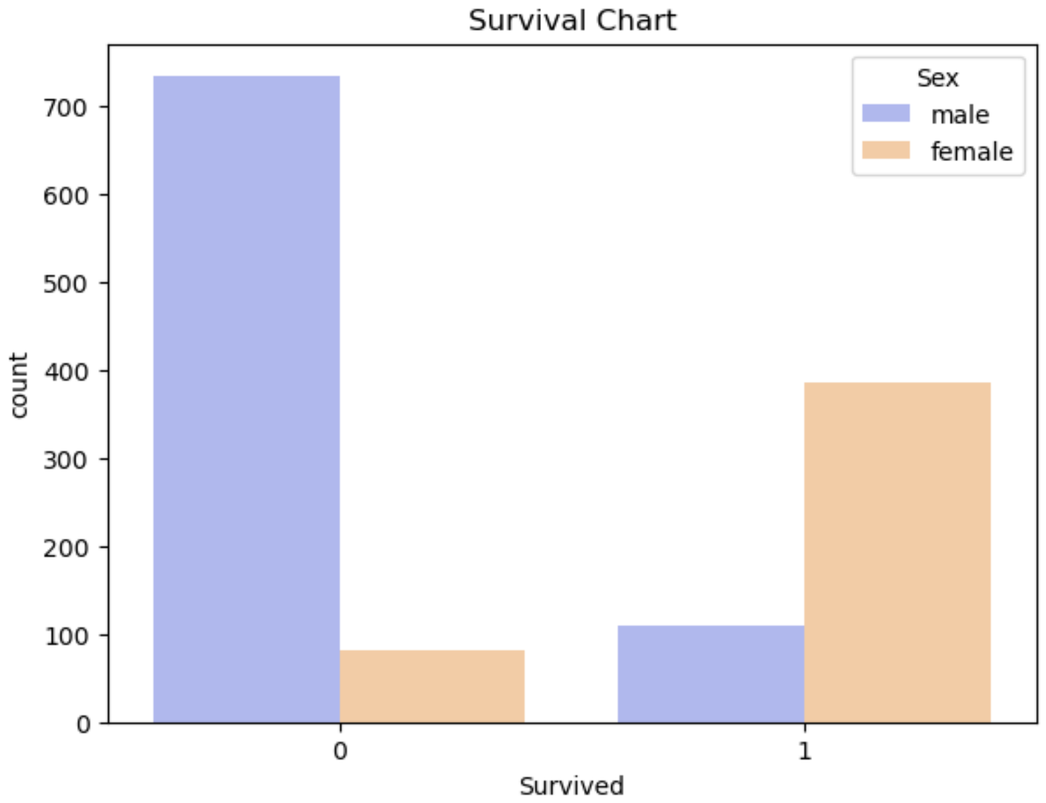
Se realiza una limpieza y normalización de nuestros datos.

```
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  1309 non-null   int64  
 1   Survived     1309 non-null   int64  
 2   Pclass       1309 non-null   int64  
 3   Name         1309 non-null   object 
 4   Sex          1309 non-null   object 
 5   Age          1309 non-null   float64
 6   SibSp        1309 non-null   int64  
 7   Parch        1309 non-null   float64
 8   Ticket       1309 non-null   object 
 9   Fare         1309 non-null   float64
 10  Cabin        1068 non-null   object 
 11  Embarked     270 non-null    object 
dtypes: float64(3), int64(4), object(5)
memory usage: 122.8+ KB
```

Tipos de datos
(Creación propia, 2023).

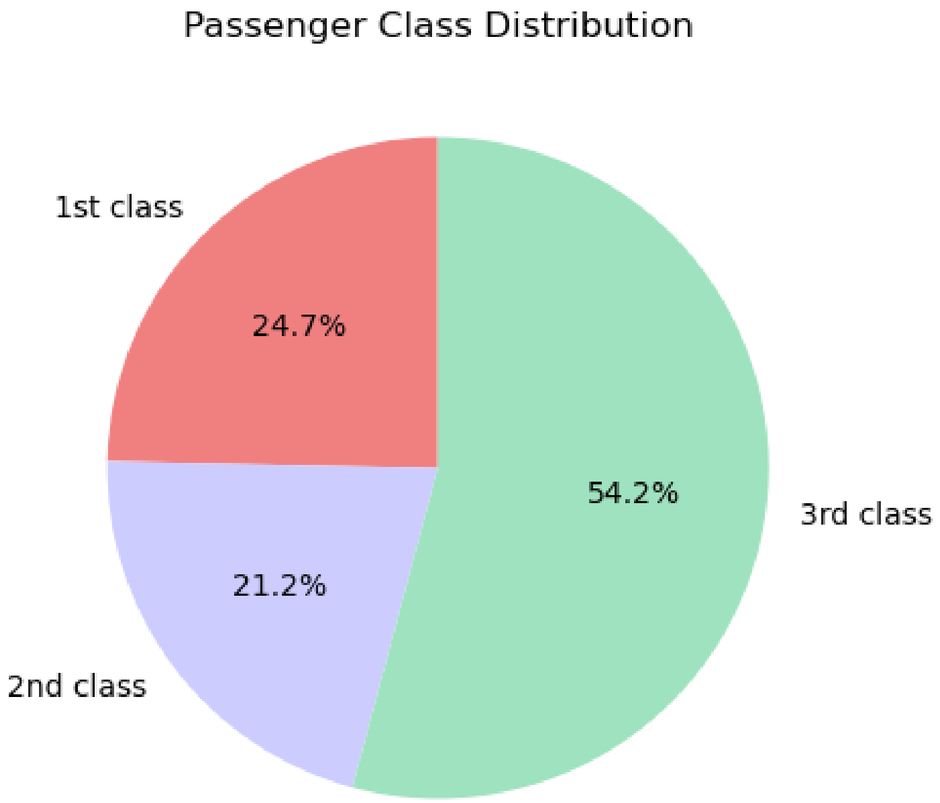
Antecedentes





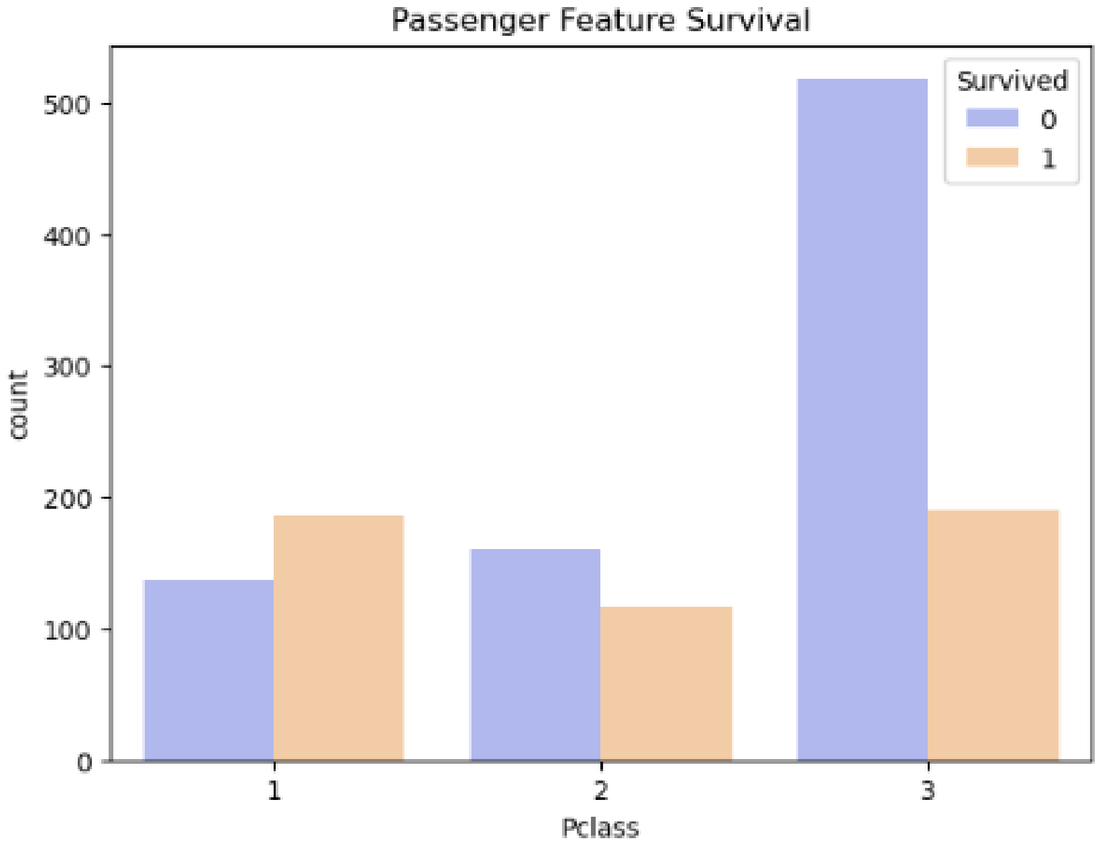
Porcentaje de supervivencia
(Creación propia, 2023).

Resultados



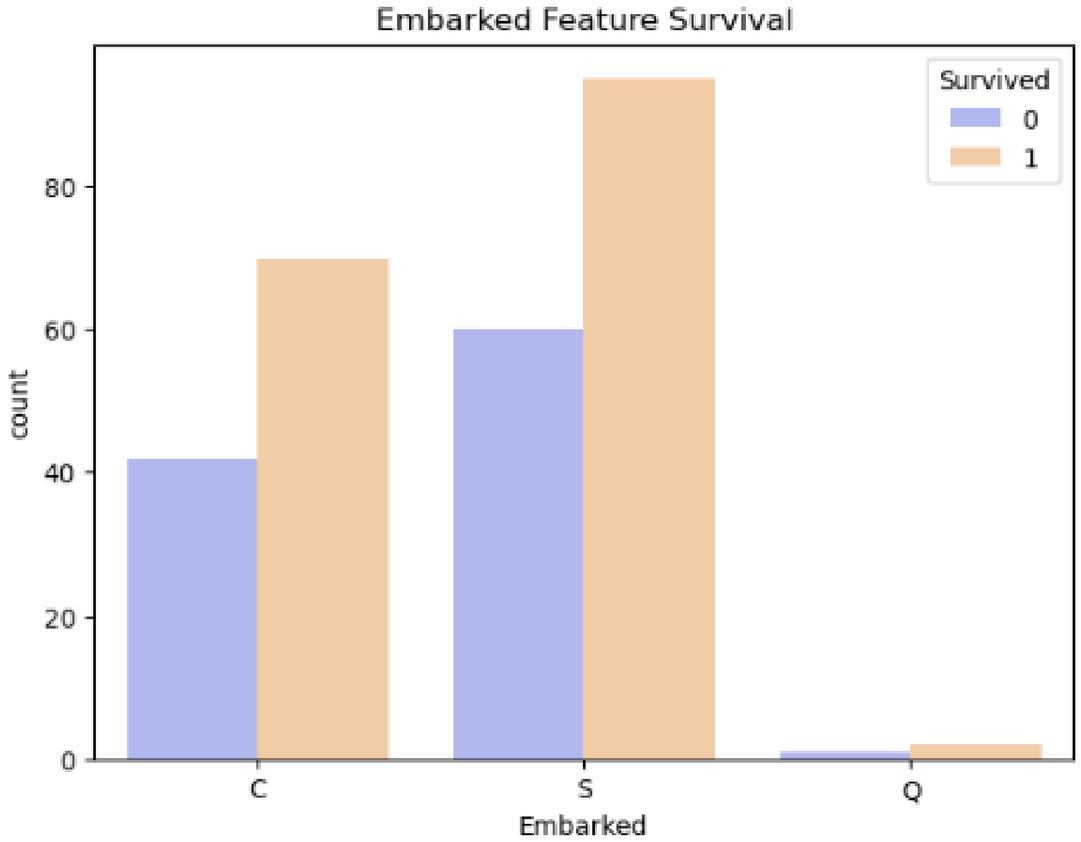
Distribución de clases en los pasajeros
(Creación propia, 2023)

Resultados



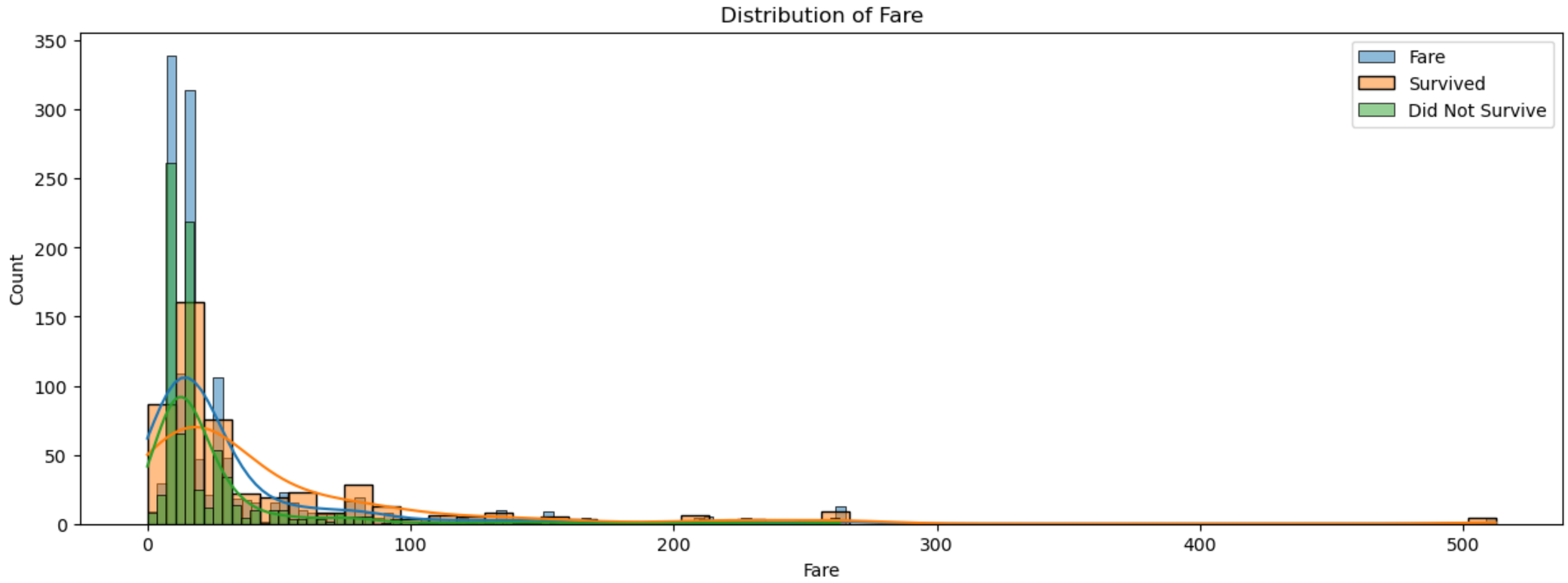
Supervivencia por clase
(Creación propia, 2023)

Resultados



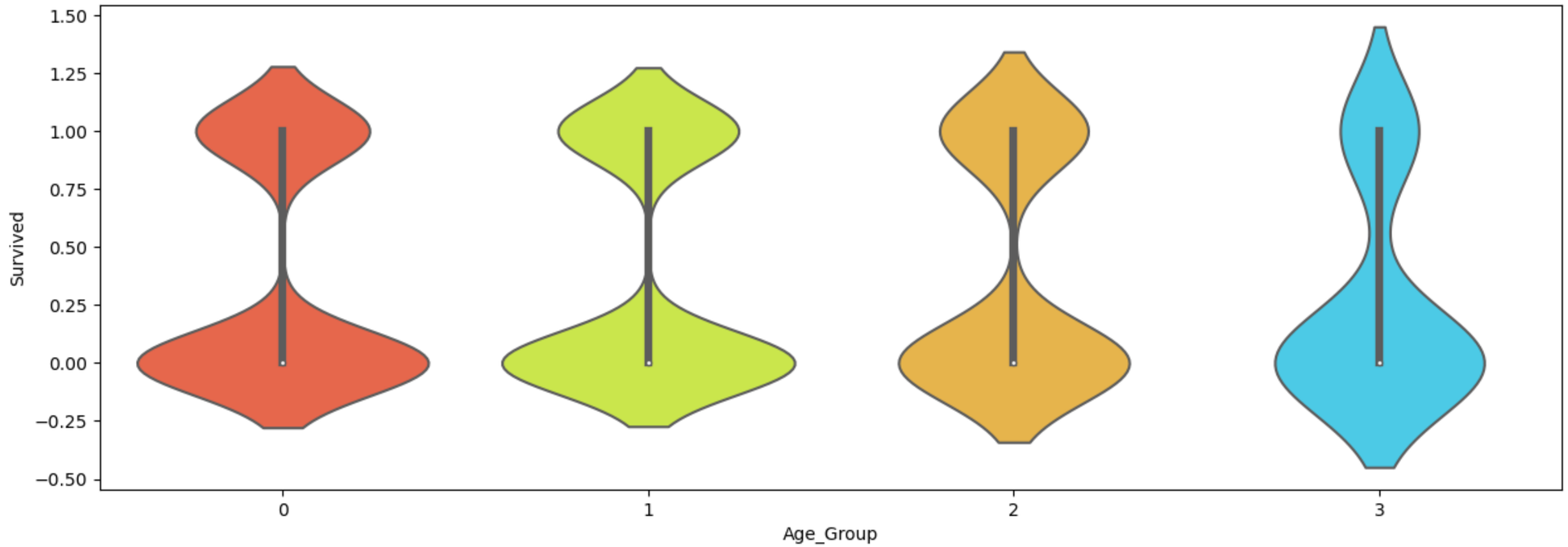
Supervivencia por embarcado
(Creación propia, 2023)

Resultados



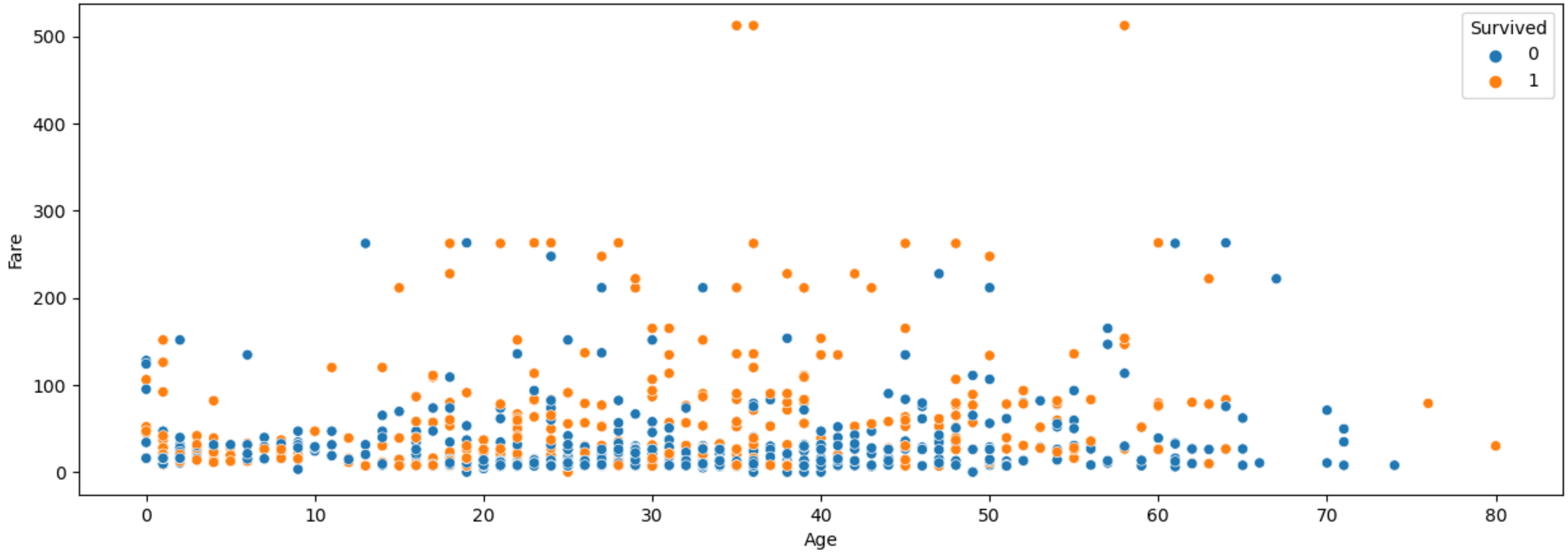
Distribución por costo de boleto
(Creación propia, 2023)

Resultados



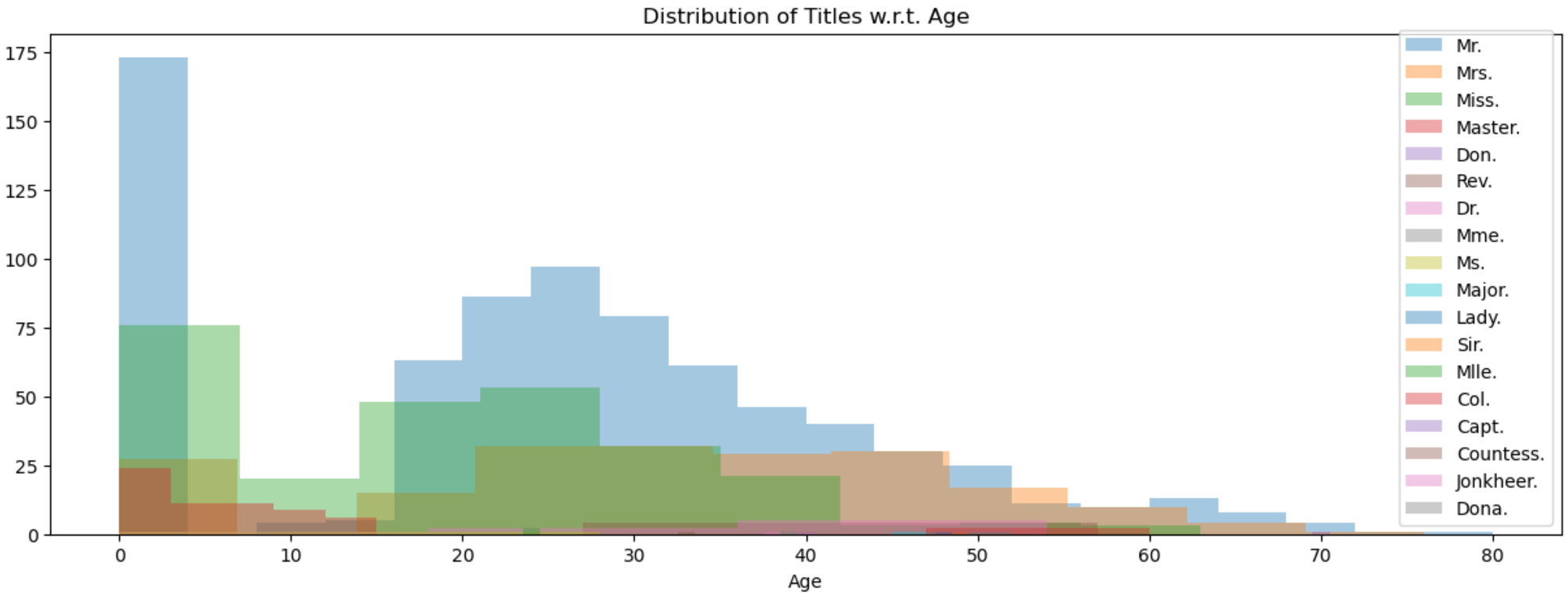
Distribución por grupo de edad
(Creación propia, 2023)

Resultados



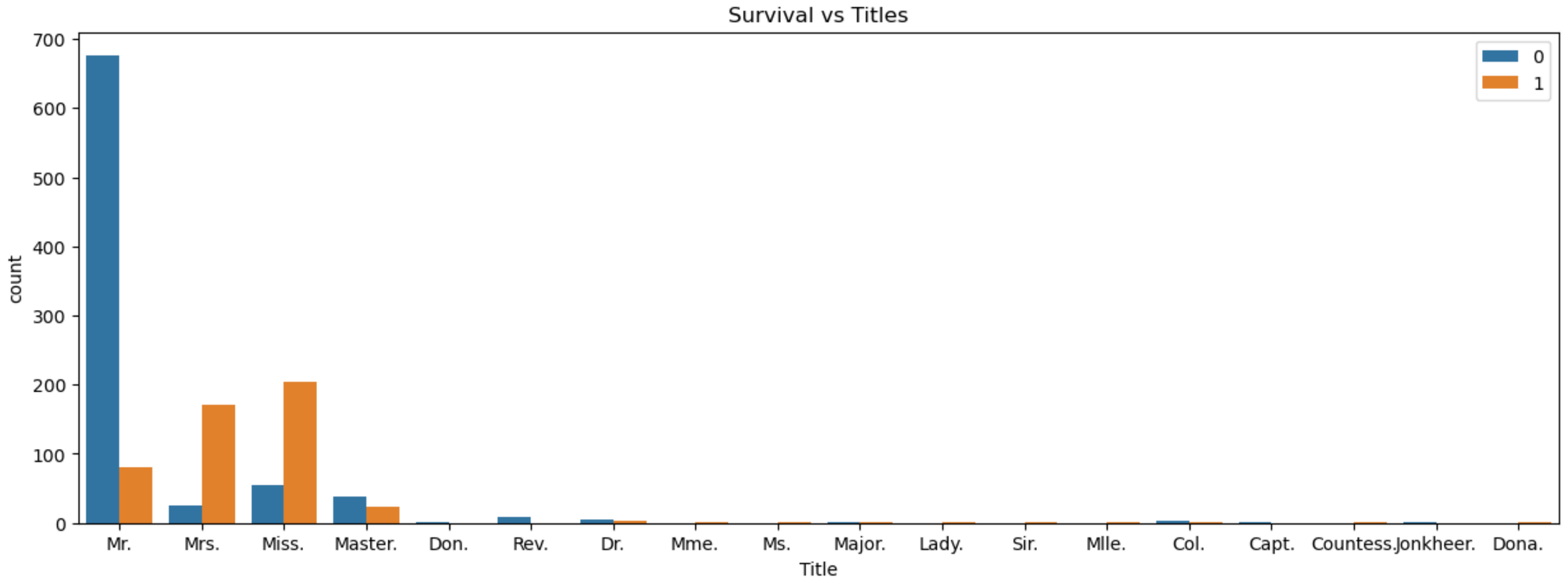
Distribución por edad y costo de boleto en relación con la supervivencia
(Creación propia, 2023)

Resultados



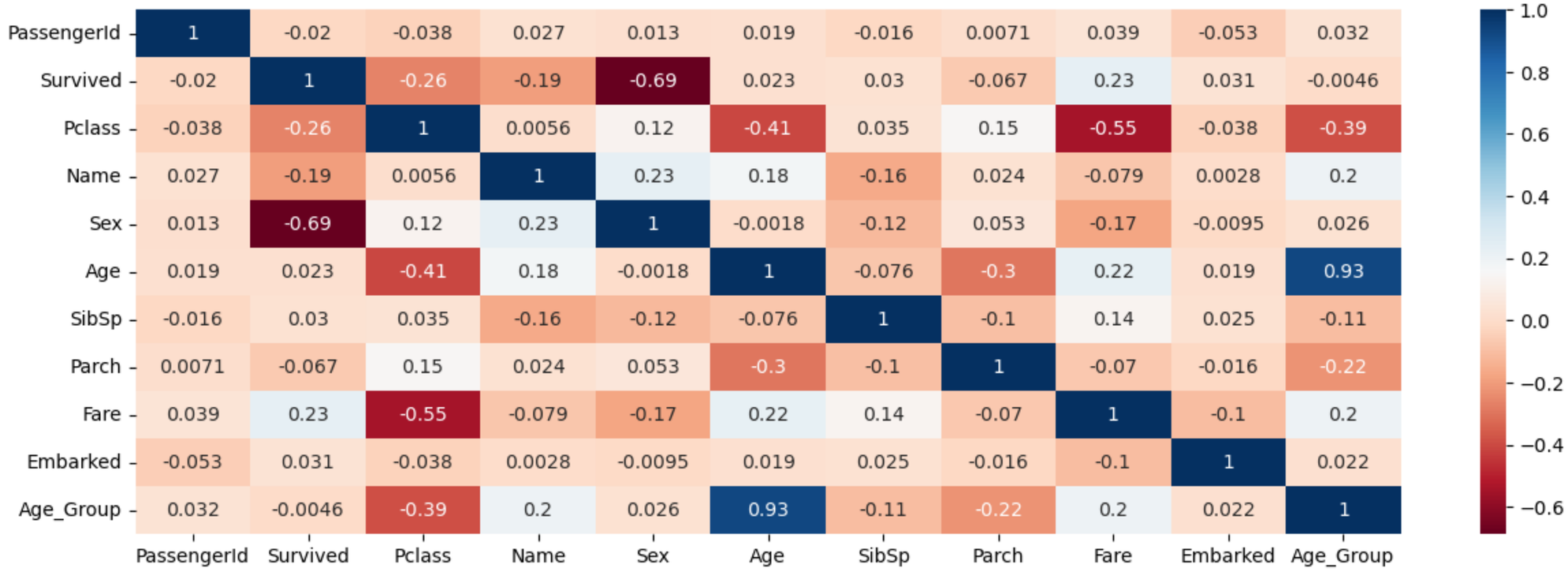
Distribución por titulo en relación con la edad
(Creación propia, 2023)

Resultados



Distribución por titulo en relación con la supervivencia
(Creación propia, 2023)

Resultados



Pearson Correlation heatmap
(Creación propia, 2023)

Resultados

	Survived	Pclass	Sex	Age	Fare
0	0	3	1	22.00	7.25
1	1	1	0	38.00	71.28
2	1	3	0	26.00	7.92
3	1	1	0	35.00	53.10
4	0	3	1	35.00	8.05

Datos después de limpieza
(Creación propia, 2023)

	Column	Score
3	Fare	5424.52
1	Sex	220.82
0	Pclass	28.04
2	Age	8.46

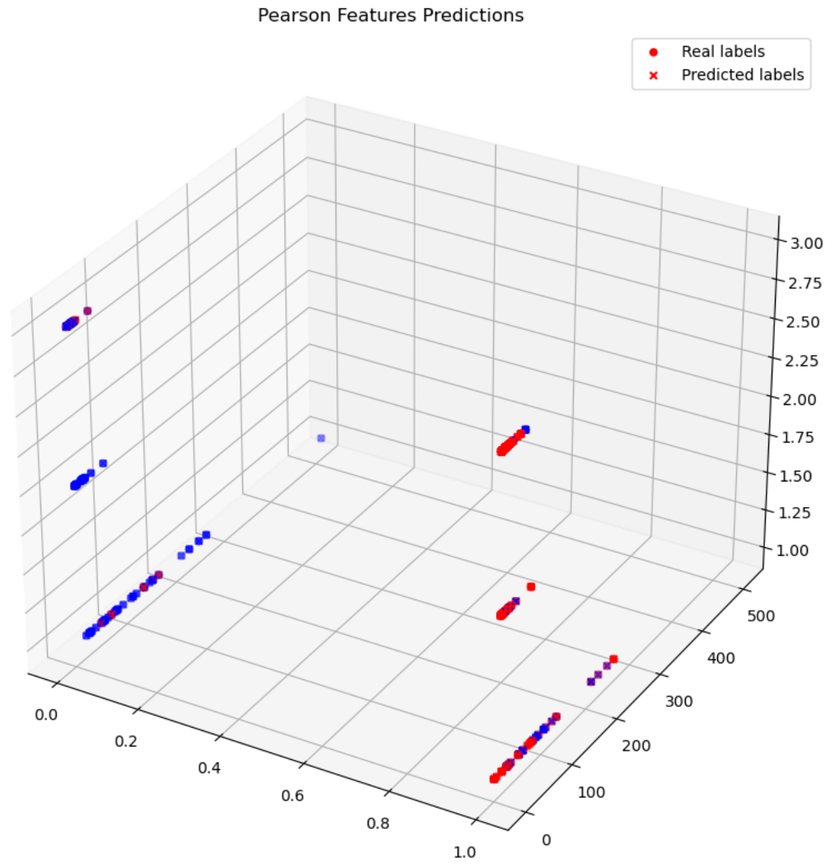
Chi-squared
(Creación propia, 2023)

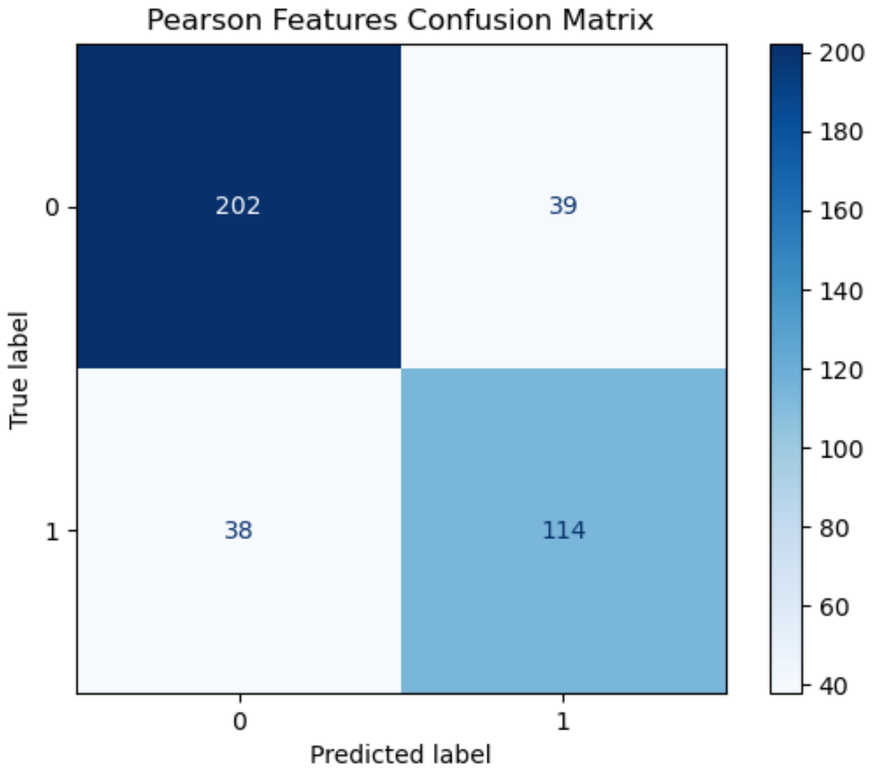
Resultados


```python
class KNNClassifier:

    def __init__(self, k=3, distance_metric="euclidean"):
        self.k = k
        self.distance_metric = distance_metric
        self.X_train = None
        self.Y_train = None

    def fit(self, X_train, Y_train):
        self.X_train = X_train
        self.Y_train = Y_train

    def predict(self, X_test):
        predictions = []
        for x_test in X_test:
            if self.distance_metric == "euclidean":
                distances = np.linalg.norm(self.X_train - x_test, axis=1)
            elif self.distance_metric == "manhattan":
                distances = np.sum(np.abs(self.X_train - x_test), axis=1)

            sorted_indexes = distances.argsort()[:self.k]
            sorted_labels = self.Y_train[sorted_indexes]
            predicted_label = np.bincount(sorted_labels).argmax()
            predictions.append(predicted_label)
        return np.array(predictions)
```

Código de KNN
(Creación propia, 2023)

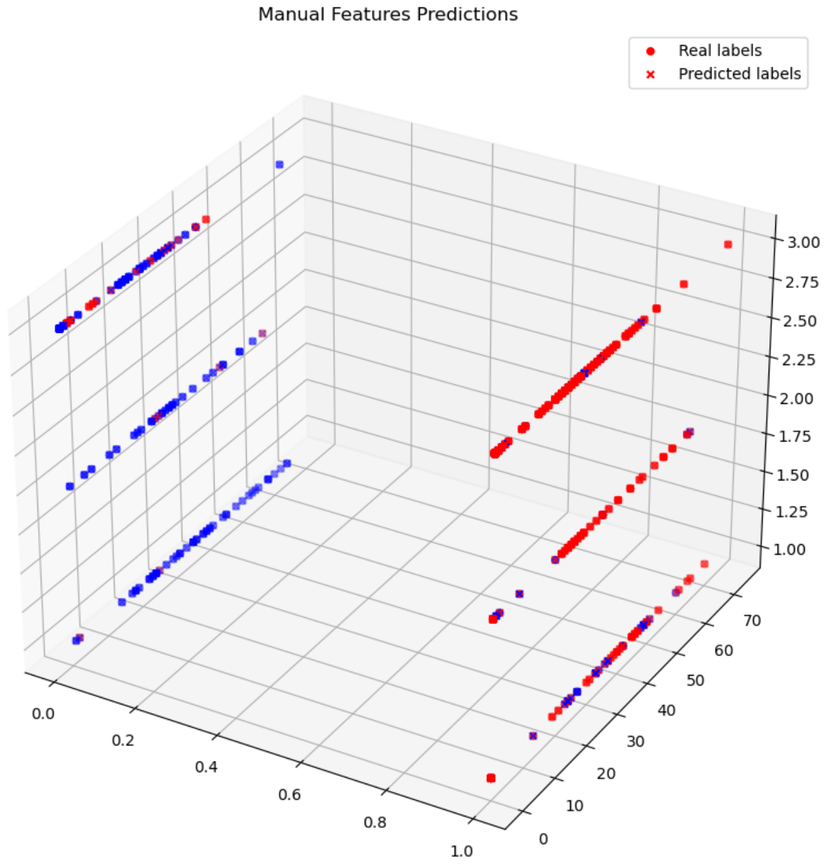
Resultados

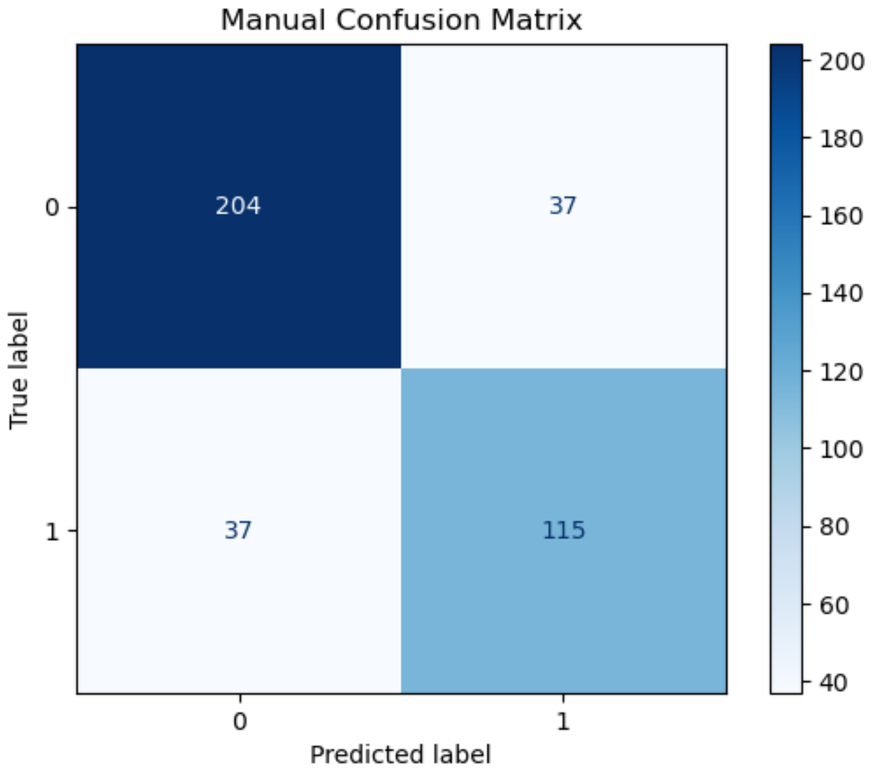




Predicción de por selección de Pearson
(Creación propia, 2023)

Resultados





Predicción de por pruebas de Chi-squared
(Creación propia, 2023)

Conclusiones

KNN puede servir como modelo de referencia para la comparación con algoritmos de aprendizaje automático más avanzados, ya que proporciona un punto de partida para el modelado y ayuda a establecer un punto de referencia para la evaluación del rendimiento.


Gracias