

Titanic Challenge *

Qzer - 2023

* Amine, Carlo, Davide, Enrico, Federico, Giulio

Il Titanic. Qualche numero

Il **Titanic** è stato un transatlantico naufragato il 15 aprile 1912.
I passeggeri del Titanic erano teoricamente **2224**, di cui:

- 324 in prima classe;
- 284 in seconda;
- 709 in terza;
- 906 membri dell'equipaggio

Stando ai numeri ufficiali, nel disastro persero la vita ben 1502,
67,54%

Challenge

Costruire un modello predittivo che risponda alla domanda: **“che tipo di persone avevano maggiori probabilità di sopravvivere?”**

La **Titanic Challenge** è un esempio classico di problema di classificazione che viene spesso utilizzato per mostrare come funzionano le reti neurali



Dati

In questa competizione avremo accesso a due set di dati simili che includono informazioni sui passeggeri come:

- nome;
- età;
- genere;
- classe socio-economica;
- ecc.

How ?

1. Data selection strategy;
2. Data preparation;
3. Algorithm training and testing

Data selection strategy

Train.csv

```
training_set = pd.read_csv('/kaggle/input/titanic/train.csv')
```

#	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

Test.csv

```
testing_set = pd.read_csv('/kaggle/input/titanic/test.csv')
```

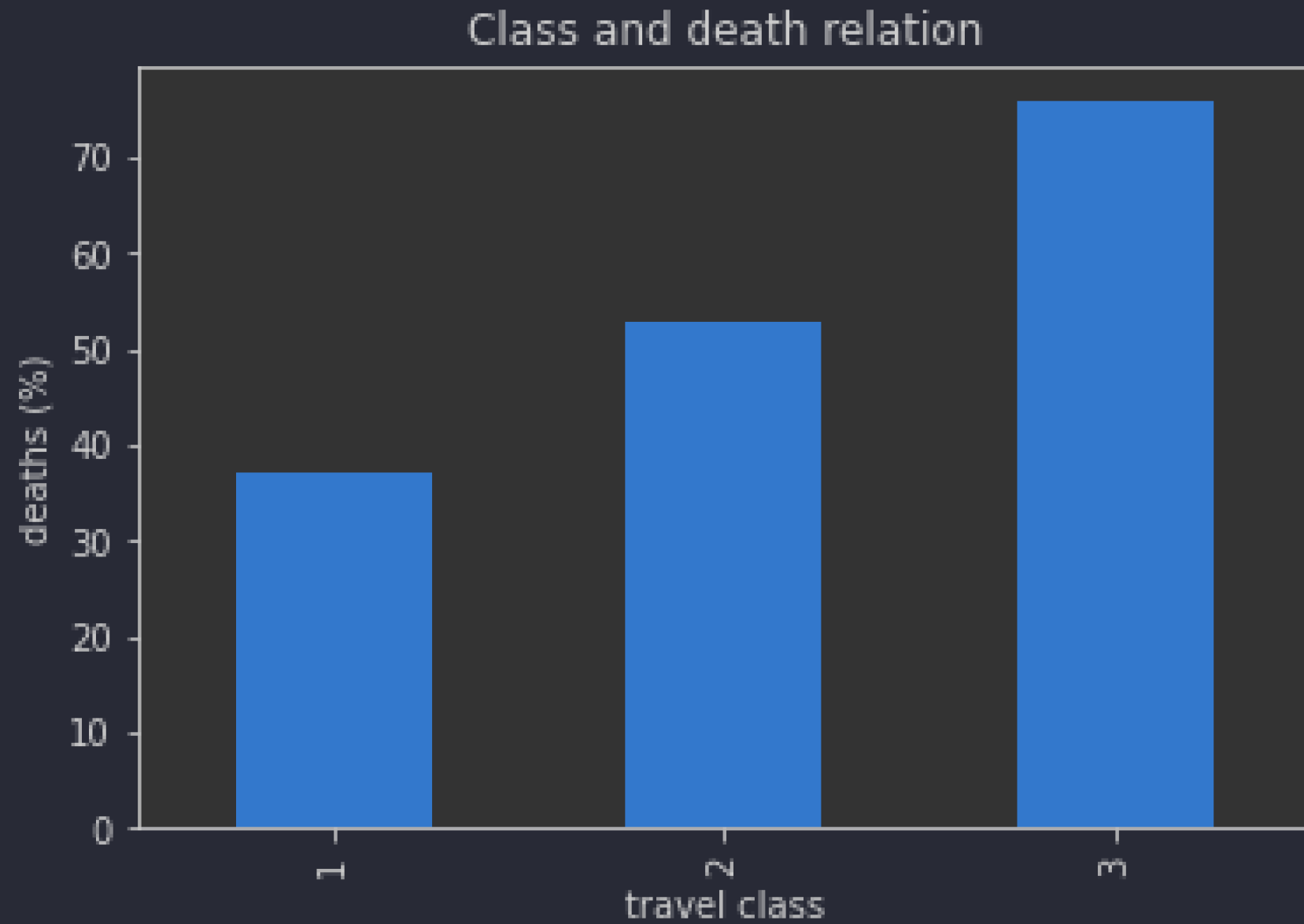
#	Column	Non-Null	Count	Dtype
#	-----	-----	-----	-----
0	PassengerId	418	non-null	int64
1	Pclass	418	non-null	int64
2	Name	418	non-null	object
3	Sex	418	non-null	object
4	Age	332	non-null	float64
5	SibSp	418	non-null	int64
6	Parch	418	non-null	int64
7	Ticket	418	non-null	object
8	Fare	417	non-null	float64
9	Cabin	91	non-null	object
10	Embarked	418	non-null	object

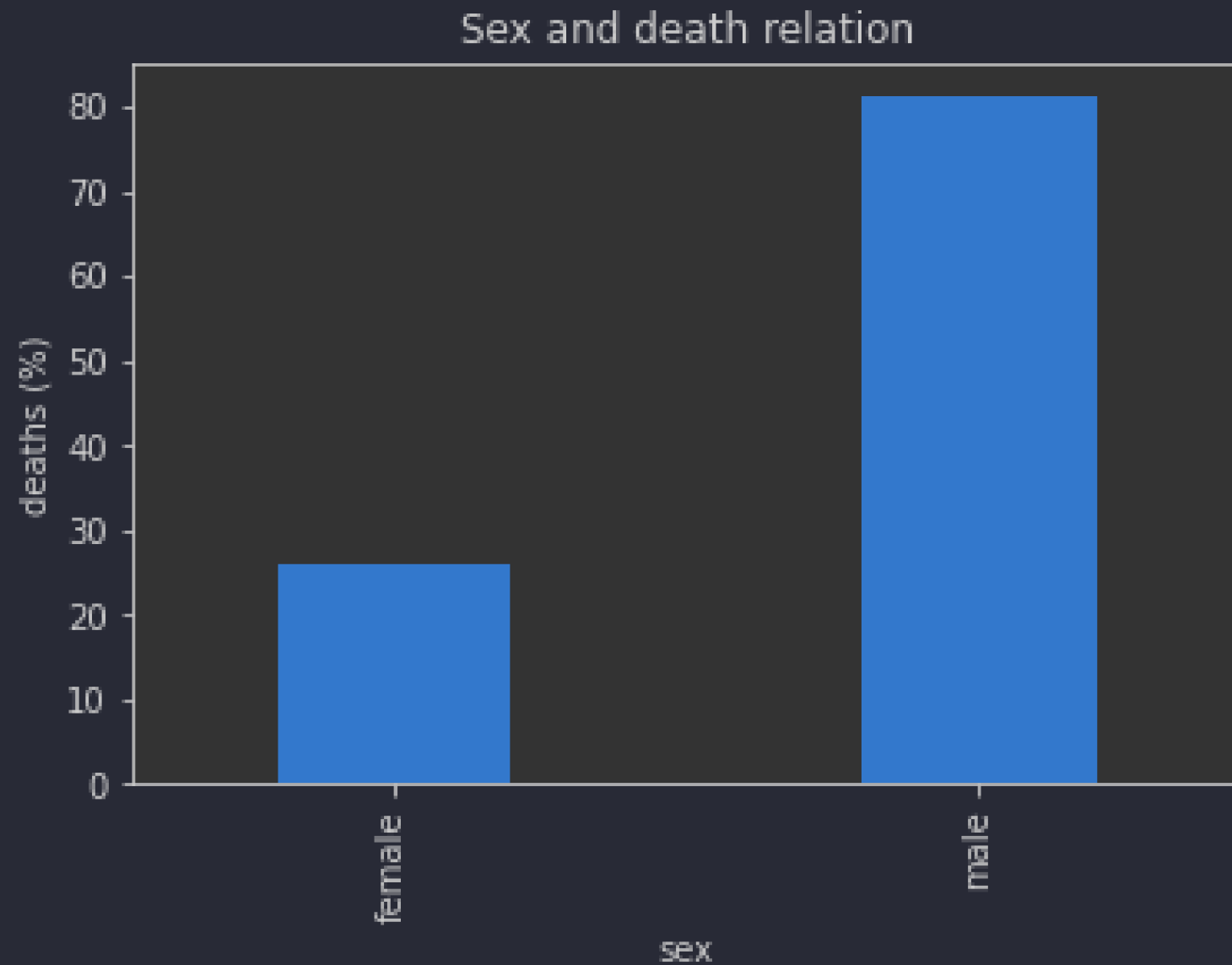
Data selection

```
RangeIndex: 891 entries, 0 to 890
#    Column      Non-Null Count  Dtype
#    -----  -
0    Pclass      891 non-null    int64
1    Sex          891 non-null    object
2    Age          714 non-null    float64
3    Parch        891 non-null    int64
4    SibSp         891 non-null    int64
5    Embarked     889 non-null    object
6    Survived     891 non-null    int64
```

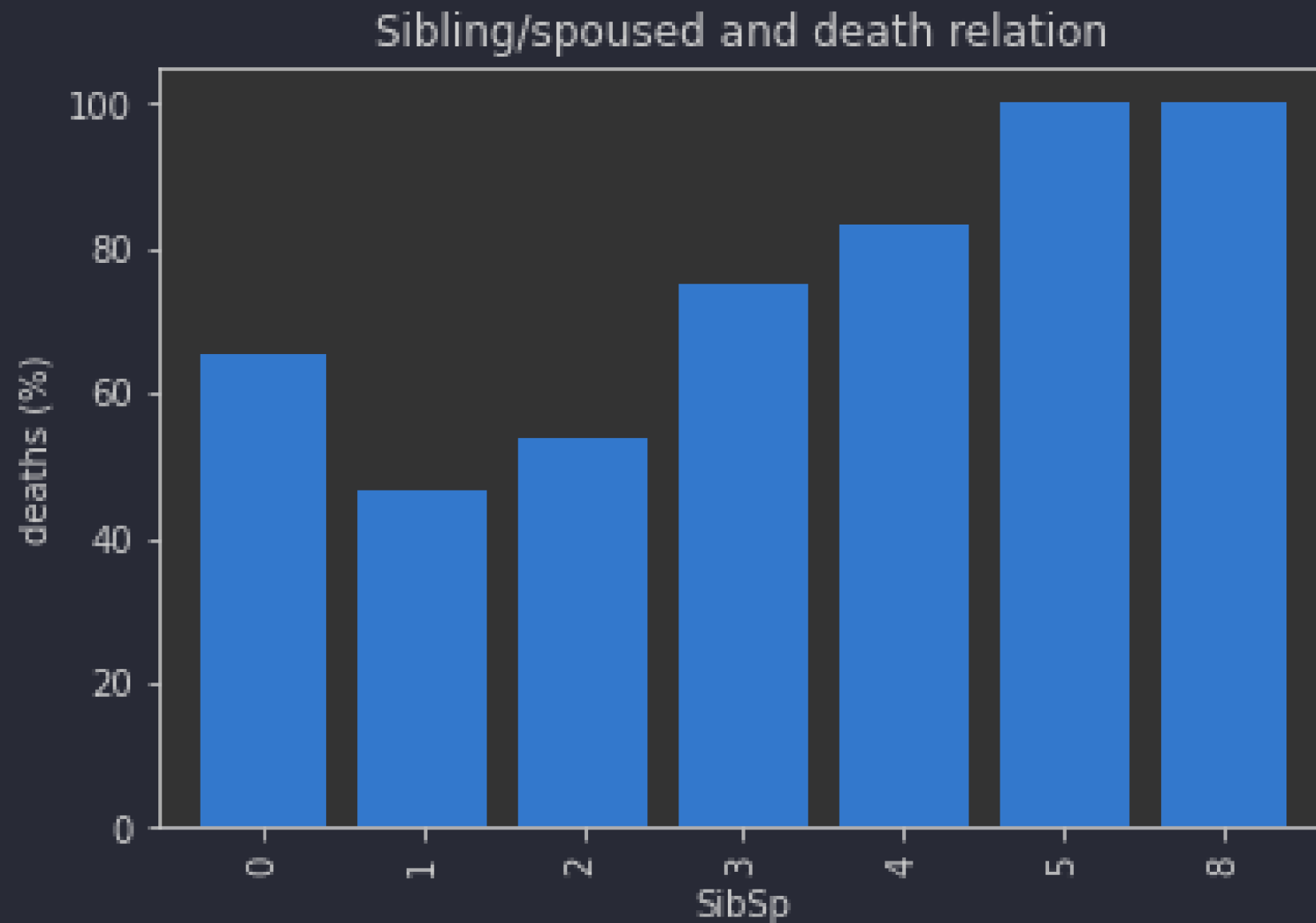
Queste le colonne trattenute dal file `train.csv`

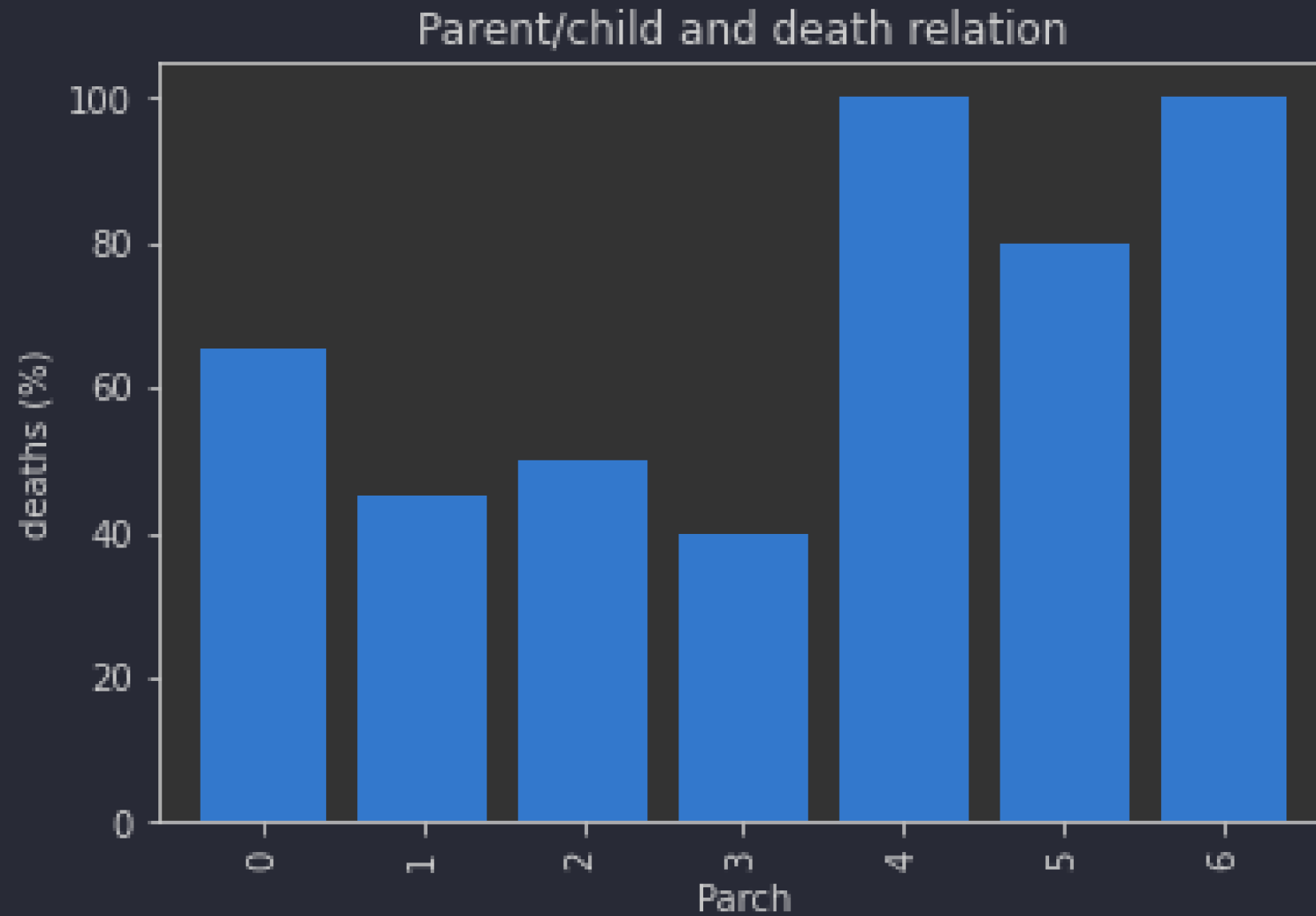
Correlazioni

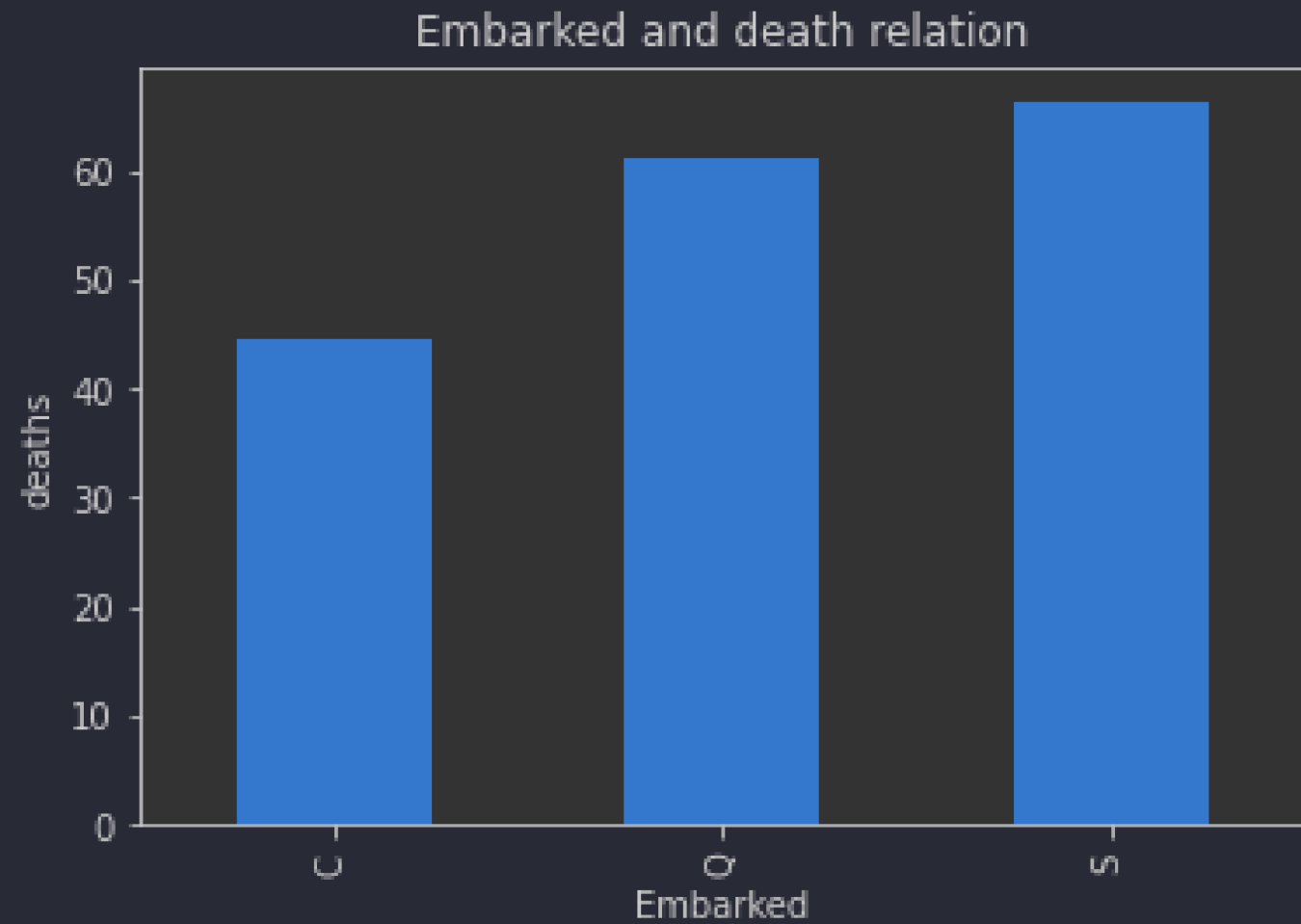












CCherbourg-QQueenstown-SSouthampton

Data preparation

Encoding del DataSet

Convertire i **maschi** nel valore **0** e le **femmine** nel valore **1**

Sostituire i valori **NaN** della colonna Age con **l'età media**, ossia 30 anni

Nella colonna **Embarked** sostituire coi numeri **0**, **1** e **2** le lettere **C**(herbourg), **Q**(ueenstown) e **S**(outhampton):

```
# Transforms from numbers to strings
df["Embarked"] = df.Embarked.map({"C":0, "Q":1, "S":2})
```

Encoding del DataSet

Dividere la colonna Age in **Child**, **Adult** e **Elderly**

```
child_list = df['Age'].apply(lambda x: 1 if x < 18 else 0)
df.insert(4, "Child", child_list, True)

adult_list = df['Age'].apply(lambda x: 1 if x >= 18 and x < 50 else 0)
df.insert(5, "Adult", adult_list, True)

elderly_list = df['Age'].apply(lambda x: 1 if x > 50 else 0)
df.insert(6, "Elderly", elderly_list, True)
```

DataSet finale

PassengerId	Pclass	Sex	Child	Adult	Elderly	Parch	SibSp	Embarked
0	892	3	0	0	1	0	0	1
1	893	3	1	0	1	0	0	1
2	894	2	0	0	0	1	0	0
3	895	3	0	0	1	0	0	0
4	896	3	1	0	1	0	1	1
...
413	1305	3	0	0	1	0	0	2
414	1306	1	1	0	1	0	0	0
415	1307	3	0	0	1	0	0	2
416	1308	3	0	0	1	0	0	2
417	1309	3	0	0	1	0	1	0

Algorithm training and testing

Algoritmo

Multi-layer Perceptron classifier (MLP).

L'algoritmo **MLP** è un metodo per addestrare le reti neurali multistrato. Consiste nel modificare i pesi delle connessioni tra i neuroni della rete neurale in modo da ridurre l'errore tra l'output della rete neurale e l'output desiderato. L'algoritmo viene ripetuto finché l'errore non raggiunge un livello accettabile.

Risultato

L'accuratezza dell'algoritmo con il training suddiviso in due parti è del **78.77**

Utilizzando il training set già fornito da Kaggle è del **76.5**

PassengerId		Survived
0	892	0
1	893	0
2	894	0
3	895	0
4	896	0
...
413	1305	0
414	1306	1
415	1307	0
416	1308	0
417	1309	1

Summary

1. Data selection strategy;
2. Data preparation;
3. Algorithm training and testing -> 90% accuracy?

Normalizzazione

```
from sklearn import preprocessing  
scaler = preprocessing.StandardScaler().fit(X_Train)
```

Github

Per maggiori informazioni

[TitanicKaggle](#)



Grazie