Exercício Titanic Kaggle

September 19, 2022

1 Exercício Titanic Kaggle

Felipe Antonio Brito de Oliveira Aluvino

1.1 Importações Gerais

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
```

1.2 Importações de Pré-Processamento

[2]:
```
from sklearn.preprocessing import OneHotEncoder, LabelEncoder, label_binarize
```

1.3 Importações Machine Learning

[3]:
```
import catboost
from sklearn.model_selection import train_test_split
from sklearn import model_selection, tree, preprocessing, metrics, linear_model
from sklearn.svm import LinearSVC
from sklearn.ensemble import GradientBoostingClassifier, RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LinearRegression, LogisticRegression,␣
↪SGDClassifier
from sklearn.tree import DecisionTreeClassifier
from catboost import CatBoostClassifier, Pool, cv
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
```

1.4 Importando os arquivos necessarios

[4]:
```
train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')
```

[5]:
```
train
```

[5]:
```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           …         …     …
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3

                                                  Name     Sex   Age  SibSp  \
0                              Braund, Mr. Owen Harris    male  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                               Heikkinen, Miss. Laina  female  26.0      0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                             Allen, Mr. William Henry    male  35.0      0
..                                                 …     …   …    …
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
889                              Behr, Mr. Karl Howell    male  26.0      0
890                                Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     …               …      …   …      …
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

[6]:
```python
train.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
```

[7]:
```python
train.isnull().sum()
```

[7]:
```
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
```

[8]:
```python
test
```

[8]:
```
     PassengerId  Pclass                                          Name  \
0            892       3                              Kelly, Mr. James
1            893       3              Wilkes, Mrs. James (Ellen Needs)
2            894       2                     Myles, Mr. Thomas Francis
3            895       3                              Wirz, Mr. Albert
4            896       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)
..           …     …                                             …
413         1305       3                            Spector, Mr. Woolf
```

```
414         1306       1                  Oliva y Ocana, Dona. Fermina
415         1307       3                  Saether, Mr. Simon Sivertsen
416         1308       3                           Ware, Mr. Frederick
417         1309       3                      Peter, Master. Michael J

        Sex   Age  SibSp  Parch              Ticket      Fare Cabin Embarked
0      male  34.5      0      0              330911    7.8292   NaN        Q
1    female  47.0      1      0              363272    7.0000   NaN        S
2      male  62.0      0      0              240276    9.6875   NaN        Q
3      male  27.0      0      0              315154    8.6625   NaN        S
4    female  22.0      1      1             3101298   12.2875   NaN        S
..      …    …    …    …                 …       …    …      …
413    male   NaN      0      0           A.5. 3236    8.0500   NaN        S
414  female  39.0      0      0            PC 17758  108.9000  C105        C
415    male  38.5      0      0  SOTON/O.Q. 3101262    7.2500   NaN        S
416    male   NaN      0      0              359309    8.0500   NaN        S
417    male   NaN      1      1                2668   22.3583   NaN        C

[418 rows x 11 columns]
```

[9]:
```
test.isnull().sum()
```

[9]:
```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

[10]:
```
#criando um DF que será enviado para o Kaggle
passengerID = test['PassengerId']

#criando um DF com o teste e o treino para tratar os dados mais rapidamente
df_titanic = pd.concat([train, test], ignore_index=True)
```

[11]:
```
df_titanic
```

[11]:
```
      PassengerId  Survived  Pclass  \
0               1       0.0       3
1               2       1.0       1
2               3       1.0       3
```

```
3                4       1.0       1
4                5       0.0       3
…             …       …       …
1304          1305       NaN       3
1305          1306       NaN       1
1306          1307       NaN       3
1307          1308       NaN       3
1308          1309       NaN       3

                                                   Name     Sex   Age  SibSp  \
0                               Braund, Mr. Owen Harris    male  22.0      1
1     Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                                Heikkinen, Miss. Laina  female  26.0      0
3          Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                              Allen, Mr. William Henry    male  35.0      0
…                                                   …     …     …    …
1304                                 Spector, Mr. Woolf    male   NaN      0
1305                       Oliva y Ocana, Dona. Fermina  female  39.0      0
1306                       Saether, Mr. Simon Sivertsen    male  38.5      0
1307                                Ware, Mr. Frederick    male   NaN      0
1308                           Peter, Master. Michael J    male   NaN      1

      Parch              Ticket      Fare Cabin Embarked
0         0           A/5 21171    7.2500   NaN        S
1         0            PC 17599   71.2833   C85        C
2         0    STON/O2. 3101282    7.9250   NaN        S
3         0              113803   53.1000  C123        S
4         0              373450    8.0500   NaN        S
…       …                 …       …    …       …
1304      0           A.5. 3236    8.0500   NaN        S
1305      0            PC 17758  108.9000  C105        C
1306      0  SOTON/O.Q. 3101262    7.2500   NaN        S
1307      0              359309    8.0500   NaN        S
1308      1                2668   22.3583   NaN        C

[1309 rows x 12 columns]
```

[12]:
```
#criando o índice para separar as df de treino e teste posteriormente
train_index = len(train)
test_index = len(df_titanic) - len(test)
```

[13]:
```
df_titanic.isnull().sum()
```

[13]:
```
PassengerId       0
Survived        418
Pclass            0
Name              0
```

```
Sex               0
Age             263
SibSp             0
Parch             0
Ticket            0
Fare              1
Cabin          1014
Embarked          2
dtype: int64
```

[14]:
```python
df_titanic.describe()
```

[14]:
```
       PassengerId    Survived       Pclass          Age        SibSp  \
count  1309.000000  891.000000  1309.000000  1046.000000  1309.000000
mean    655.000000    0.383838     2.294882    29.881138     0.498854
std     378.020061    0.486592     0.837836    14.413493     1.041658
min       1.000000    0.000000     1.000000     0.170000     0.000000
25%     328.000000    0.000000     2.000000    21.000000     0.000000
50%     655.000000    0.000000     3.000000    28.000000     0.000000
75%     982.000000    1.000000     3.000000    39.000000     1.000000
max    1309.000000    1.000000     3.000000    80.000000     8.000000

             Parch         Fare
count  1309.000000  1308.000000
mean      0.385027    33.295479
std       0.865560    51.758668
min       0.000000     0.000000
25%       0.000000     7.895800
50%       0.000000    14.454200
75%       0.000000    31.275000
max       9.000000   512.329200
```

[15]:
```python
#criando um df que iremos tratar os campos relevantes a partir da base␣
↪titanic_df
df = pd.DataFrame()
```

1.5 Tratando as coluna, usnado a 'Survived' como exemplo

[16]:
```python
# encontrando a quantidade de valores únicos em "Survived"
df_titanic['Survived'].nunique()
```

[16]: 2

[17]:
```python
# encontrando quais são os valores únicos em "Survived"
df_titanic['Survived'].unique()
```

[17]: array([0., 1., nan])

[18]:
```python
# encontrando a quantidade de valores nulos em "Survived"
df_titanic['Survived'].isnull().sum()
```

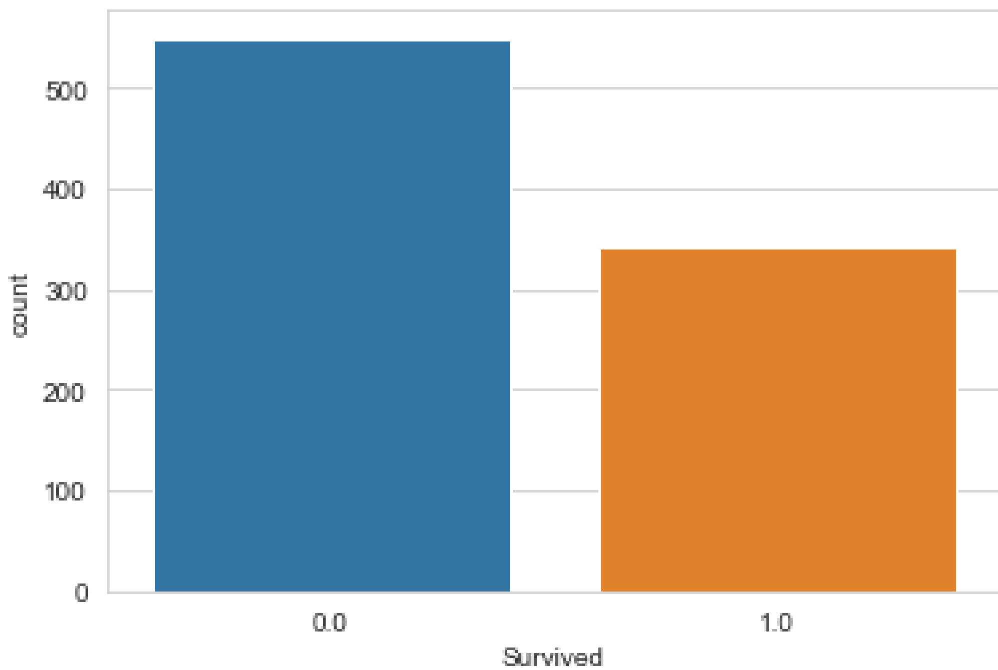
[18]: 418

[19]:
```python
# encontrando a quantidade de valores associados a cada variavel de "Survived"
df_titanic['Survived'].value_counts()
```

[19]:
```
0.0    549
1.0    342
Name: Survived, dtype: int64
```

[20]:
```python
# plotando os valores das colunas
sns.countplot(data = df_titanic, x = 'Survived')
```

[20]: <AxesSubplot:xlabel='Survived', ylabel='count'>

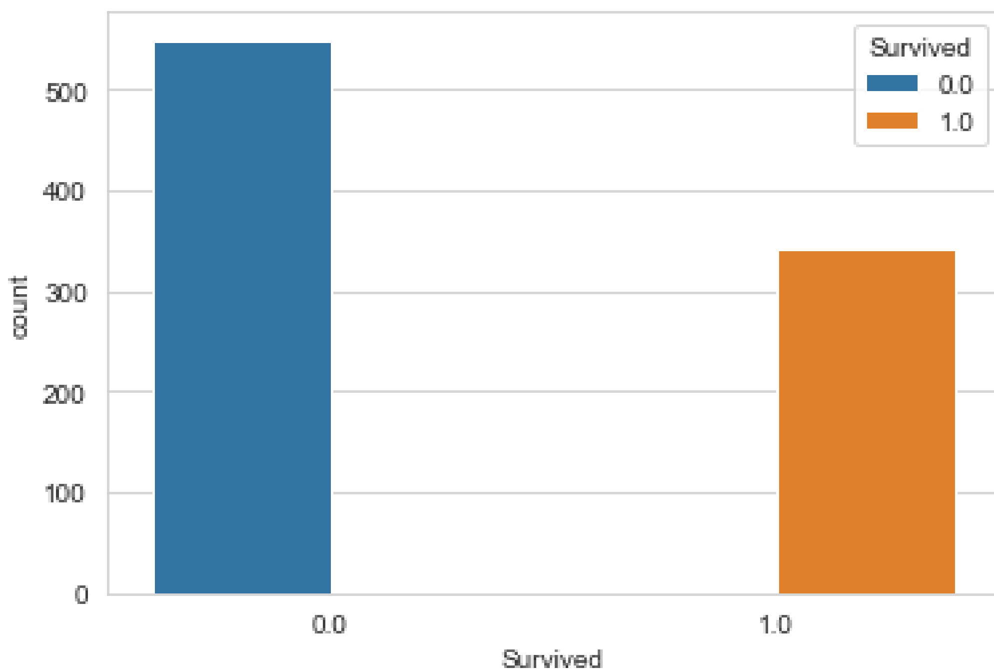


1.6 Função que traz informações sobre a coluna

[21]:
```python
#criando uma função que printa as informações sobre os valores das colunas
def df_info(data, column, count = True):
    print(f'Quantidade de valores únicos na {column}: \n{data[column].
 ↪nunique()}')
    print(f'\nQuais são os valores únicos na {column}: \n{data[column].
 ↪unique()}')
```

```
    print(f'\nQuantidade de valores nulos na {column}: \n{data[column].isnull().
↪sum()}')
    print(f'\nQuantidade por opção na {column}: \n{data[column].
↪value_counts()}')

    if count == True:
        sns.countplot(data = data, x = column, hue = 'Survived')
    else:
        sns.displot(data[column], kde = True)

df_info(df_titanic, 'Survived')
```

```
Quantidade de valores únicos na Survived:
2

Quais são os valores únicos na Survived:
[ 0.  1. nan]

Quantidade de valores nulos na Survived:
418

Quantidade por opção na Survived:
0.0    549
1.0    342
Name: Survived, dtype: int64
```



[22]:
```
df['Survived'] = df_titanic['Survived']
```

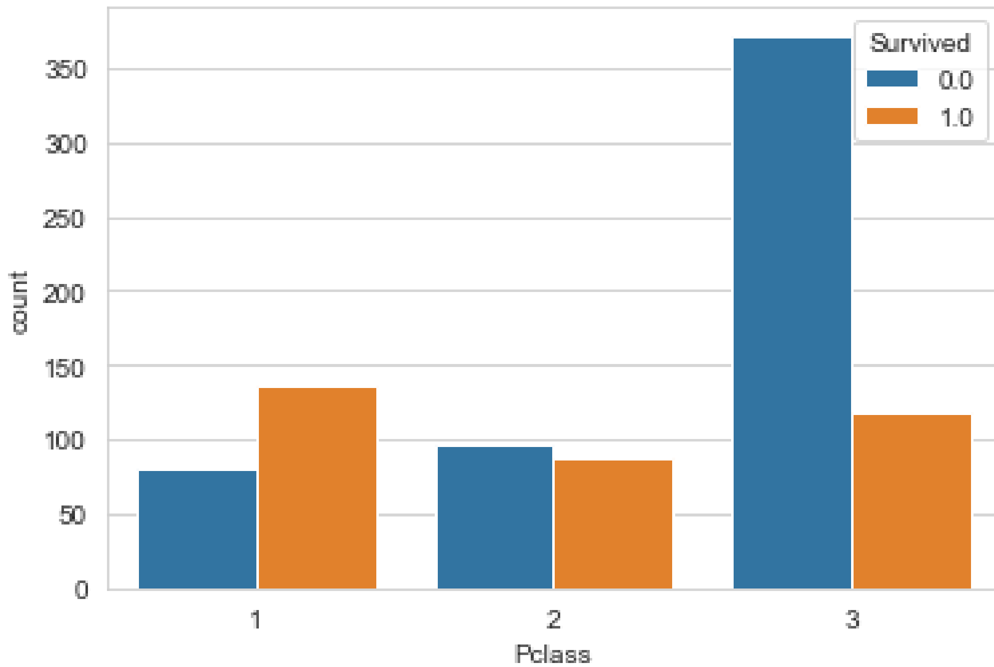
[23]:
```
df
```

[23]:
```
      Survived
0          0.0
1          1.0
2          1.0
3          1.0
4          0.0
…        …
1304       NaN
1305       NaN
1306       NaN
1307       NaN
1308       NaN

[1309 rows x 1 columns]
```

1.7 Tratando PClass

[24]:
```
df_info(df_titanic, 'Pclass')
```

```
Quantidade de valores únicos na Pclass:
3

Quais são os valores únicos na Pclass:
[3 1 2]

Quantidade de valores nulos na Pclass:
0

Quantidade por opção na Pclass:
3    709
1    323
2    277
Name: Pclass, dtype: int64
```



[25]:
```
df['Pclass'] = df_titanic['Pclass']
df
```

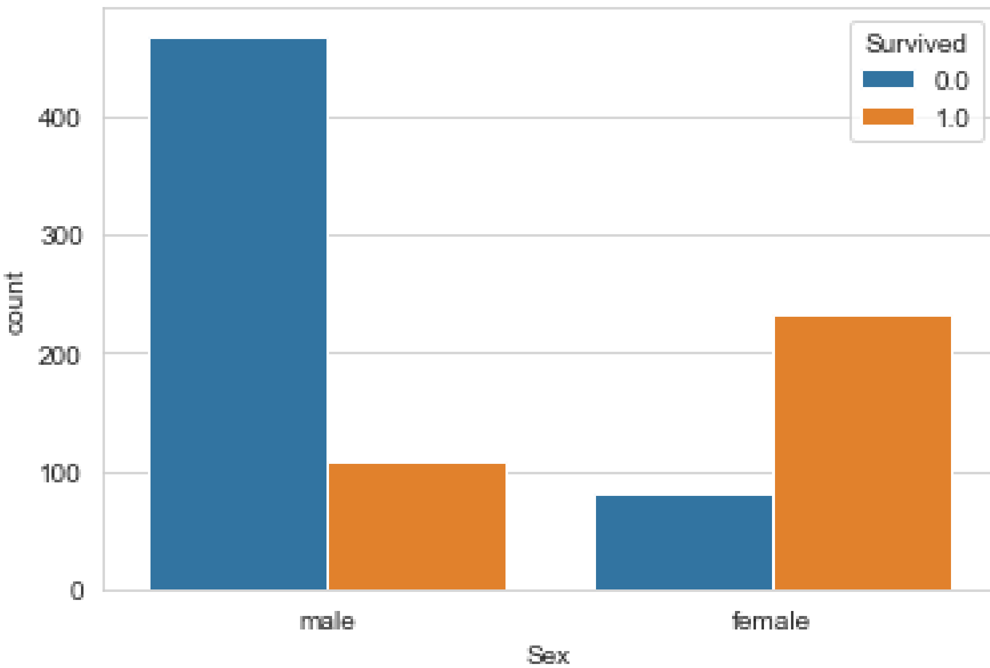
[25]:
```
      Survived  Pclass
0          0.0       3
1          1.0       1
2          1.0       3
3          1.0       1
4          0.0       3
…          …       …
1304       NaN       3
1305       NaN       1
1306       NaN       3
1307       NaN       3
1308       NaN       3

[1309 rows x 2 columns]
```

1.8 Tratando Sex

[26]:
```
df_info(df_titanic, 'Sex')
```

```
Quantidade de valores únicos na Sex:
2
```

```
Quais são os valores únicos na Sex:
['male' 'female']

Quantidade de valores nulos na Sex:
0

Quantidade por opção na Sex:
male      843
female    466
Name: Sex, dtype: int64
```



[27]:
```python
df_titanic['Sex'] = df_titanic['Sex'].replace(['male', 'female'], [0, 1])
```

[28]:
```python
df['Sex'] = df_titanic['Sex']
df
```

[28]:

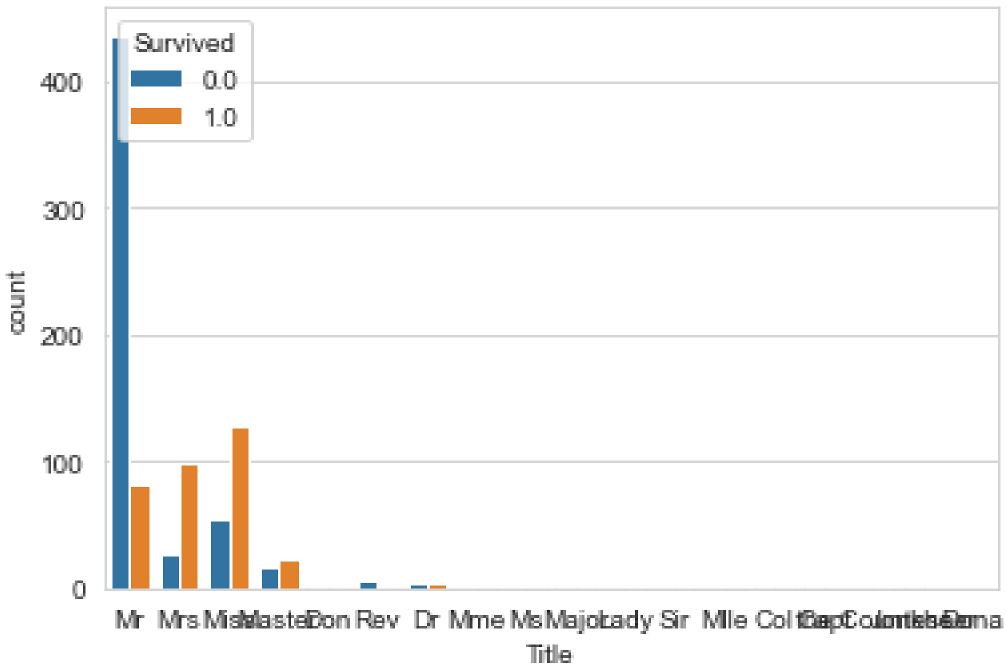
	Survived	Pclass	Sex
0	0.0	3	0
1	1.0	1	1
2	1.0	3	1
3	1.0	1	1
4	0.0	3	0
…	…	…	…
1304	NaN	3	0
1305	NaN	1	1

```
1306       NaN       3    0
1307       NaN       3    0
1308       NaN       3    0

[1309 rows x 3 columns]
```

1.9 Tratando o título

[29]:
```
df_titanic['Title'] = df_titanic['Name'].apply(lambda name: name.split(',')[1].
 ↪split('.')[0].strip())
df_info(df_titanic, 'Title')
```

```
Quantidade de valores únicos na Title:
18

Quais são os valores únicos na Title:
['Mr' 'Mrs' 'Miss' 'Master' 'Don' 'Rev' 'Dr' 'Mme' 'Ms' 'Major' 'Lady'
 'Sir' 'Mlle' 'Col' 'Capt' 'the Countess' 'Jonkheer' 'Dona']

Quantidade de valores nulos na Title:
0

Quantidade por opção na Title:
Mr              757
Miss            260
Mrs             197
Master           61
Rev               8
Dr                8
Col               4
Mlle              2
Major             2
Ms                2
Lady              1
Sir               1
Mme               1
Don               1
Capt              1
the Countess      1
Jonkheer          1
Dona              1
Name: Title, dtype: int64
```




```
[30]: df_titanic
```

```
[30]:       PassengerId  Survived  Pclass  \
      0               1       0.0       3
      1               2       1.0       1
      2               3       1.0       3
      3               4       1.0       1
      4               5       0.0       3
      …             …       …     …
      1304         1305       NaN       3
      1305         1306       NaN       1
      1306         1307       NaN       3
      1307         1308       NaN       3
      1308         1309       NaN       3

                                                       Name  Sex   Age  SibSp  \
      0                                 Braund, Mr. Owen Harris    0  22.0      1
      1      Cumings, Mrs. John Bradley (Florence Briggs Th…    1  38.0      1
      2                                  Heikkinen, Miss. Laina    1  26.0      0
      3            Futrelle, Mrs. Jacques Heath (Lily May Peel)    1  35.0      1
      4                                Allen, Mr. William Henry    0  35.0      0
      …                                                   …  …   …    …
      1304                                   Spector, Mr. Woolf    0   NaN      0
      1305                         Oliva y Ocana, Dona. Fermina    1  39.0      0
      1306                         Saether, Mr. Simon Sivertsen    0  38.5      0
```

```
1307                              Ware, Mr. Frederick    0   NaN      0
1308                         Peter, Master. Michael J    0   NaN      1

      Parch              Ticket      Fare Cabin Embarked   Title
0         0           A/5 21171    7.2500   NaN        S      Mr
1         0            PC 17599   71.2833   C85        C     Mrs
2         0    STON/O2. 3101282    7.9250   NaN        S    Miss
3         0              113803   53.1000  C123        S     Mrs
4         0              373450    8.0500   NaN        S      Mr
…      …                   …    …   …      …     …
1304      0           A.5. 3236    8.0500   NaN        S      Mr
1305      0            PC 17758  108.9000  C105        C    Dona
1306      0  SOTON/O.Q. 3101262    7.2500   NaN        S      Mr
1307      0              359309    8.0500   NaN        S      Mr
1308      1                2668   22.3583   NaN        C  Master

[1309 rows x 13 columns]
```

[31]:

```python
number_title = dict(df_titanic['Title'].value_counts())
keys_number_title = list(number_title)
keys_number_title
```

[31]:

```
['Mr',
 'Miss',
 'Mrs',
 'Master',
 'Rev',
 'Dr',
 'Col',
 'Mlle',
 'Major',
 'Ms',
 'Lady',
 'Sir',
 'Mme',
 'Don',
 'Capt',
 'the Countess',
 'Jonkheer',
 'Dona']
```

[32]:

```python
df_titanic['Title'] = [n if n in keys_number_title[0:4] else 'Person' for n in␣
 ↪df_titanic['Title']]
df_titanic
```

[32]:

```
      PassengerId  Survived  Pclass  \
0               1       0.0       3
```

```
1              2       1.0       1
2              3       1.0       3
3              4       1.0       1
4              5       0.0       3
…            …       …       …
1304        1305       NaN       3
1305        1306       NaN       1
1306        1307       NaN       3
1307        1308       NaN       3
1308        1309       NaN       3

                                                   Name  Sex   Age  SibSp  \
0                               Braund, Mr. Owen Harris    0  22.0      1
1     Cumings, Mrs. John Bradley (Florence Briggs Th…    1  38.0      1
2                                Heikkinen, Miss. Laina    1  26.0      0
3          Futrelle, Mrs. Jacques Heath (Lily May Peel)    1  35.0      1
4                              Allen, Mr. William Henry    0  35.0      0
…                                                   …   …   …    …
1304                                 Spector, Mr. Woolf    0   NaN      0
1305                       Oliva y Ocana, Dona. Fermina    1  39.0      0
1306                       Saether, Mr. Simon Sivertsen    0  38.5      0
1307                                Ware, Mr. Frederick    0   NaN      0
1308                           Peter, Master. Michael J    0   NaN      1

      Parch              Ticket      Fare Cabin Embarked   Title
0         0           A/5 21171    7.2500   NaN        S      Mr
1         0            PC 17599   71.2833   C85        C     Mrs
2         0    STON/O2. 3101282    7.9250   NaN        S    Miss
3         0              113803   53.1000  C123        S     Mrs
4         0              373450    8.0500   NaN        S      Mr
…       …                 …      …   …     …       …
1304      0          A.5. 3236    8.0500   NaN        S      Mr
1305      0            PC 17758  108.9000  C105        C  Person
1306      0  SOTON/O.Q. 3101262    7.2500   NaN        S      Mr
1307      0              359309    8.0500   NaN        S      Mr
1308      1                2668   22.3583   NaN        C  Master

[1309 rows x 13 columns]
```

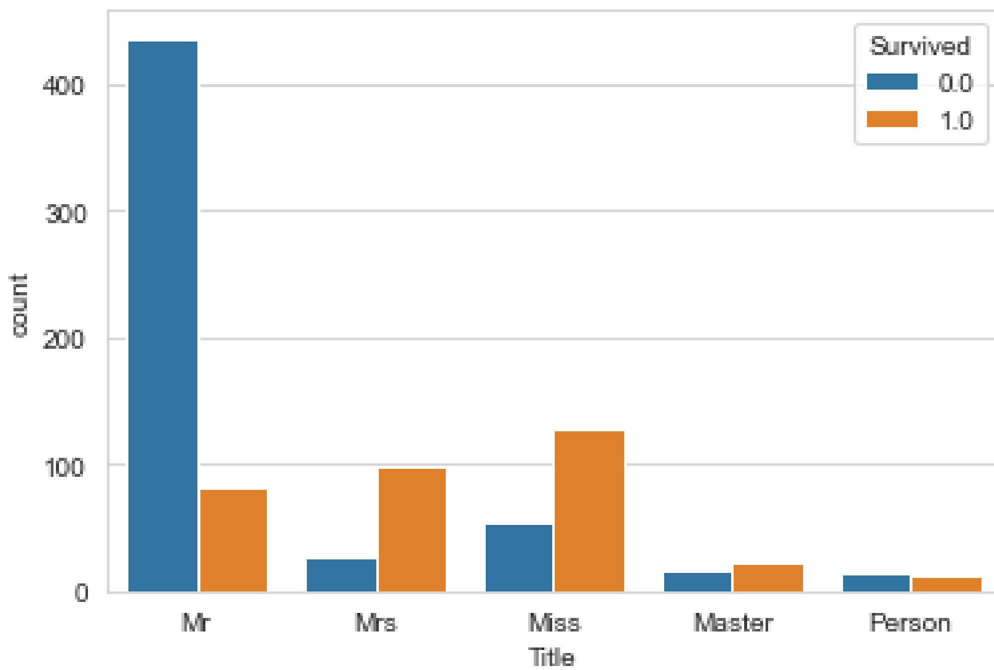
[33]:
```
number_title_actual = dict(df_titanic['Title'].value_counts())
keys_number_title_actual = list(number_title_actual)
keys_number_title_actual
```

[33]: ['Mr', 'Miss', 'Mrs', 'Master', 'Person']

[34]:
```
df['Title'] = df_titanic['Title']
```

[35]:
```
df_info(df, 'Title')
```

```
Quantidade de valores únicos na Title:
5

Quais são os valores únicos na Title:
['Mr' 'Mrs' 'Miss' 'Master' 'Person']

Quantidade de valores nulos na Title:
0

Quantidade por opção na Title:
Mr        757
Miss      260
Mrs       197
Master     61
Person     34
Name: Title, dtype: int64
```



[36]:
```
df
```

[36]:

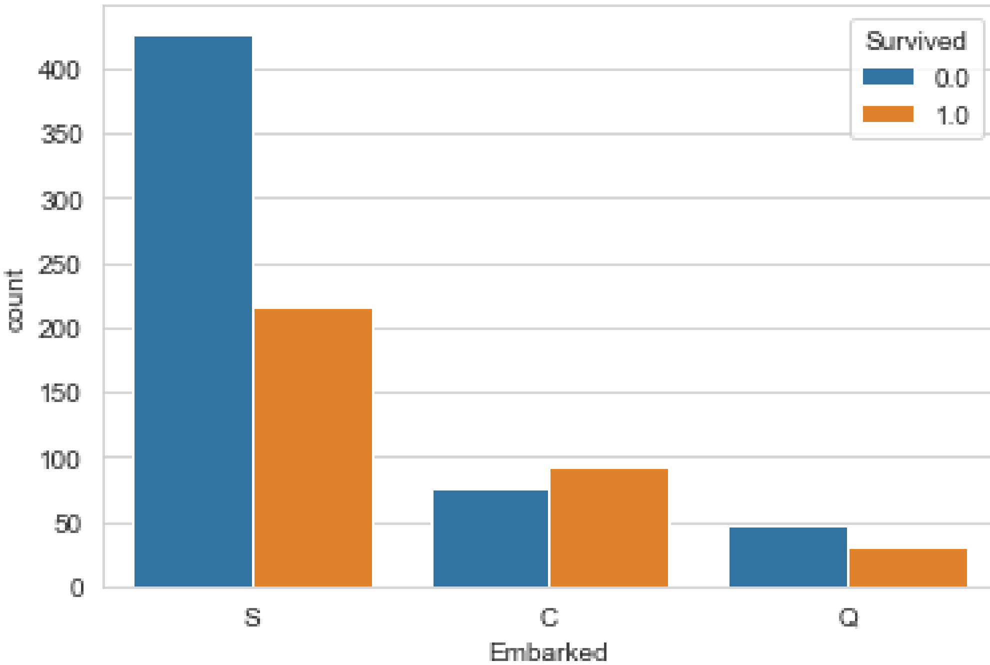
	Survived	Pclass	Sex	Title
0	0.0	3	0	Mr
1	1.0	1	1	Mrs
2	1.0	3	1	Miss

```
3           1.0        1    1      Mrs
4           0.0        3    0       Mr
…        …      …   …      …
1304        NaN        3    0       Mr
1305        NaN        1    1   Person
1306        NaN        3    0       Mr
1307        NaN        3    0       Mr
1308        NaN        3    0   Master

[1309 rows x 4 columns]
```

1.10 Tratar Embarked

[37]:
```
df_info(df_titanic, 'Embarked')
```

```
Quantidade de valores únicos na Embarked:
3

Quais são os valores únicos na Embarked:
['S' 'C' 'Q' nan]

Quantidade de valores nulos na Embarked:
2

Quantidade por opção na Embarked:
S    914
C    270
Q    123
Name: Embarked, dtype: int64
```



[38]:
```
df_titanic.loc[df_titanic['Embarked'].isnull()]
```

[38]:
```
     PassengerId  Survived  Pclass                                       Name  \
61            62       1.0       1                        Icard, Miss. Amelie
829          830       1.0       1  Stone, Mrs. George Nelson (Martha Evelyn)

     Sex   Age  SibSp  Parch  Ticket  Fare Cabin Embarked Title
61     1  38.0      0      0  113572  80.0   B28      NaN  Miss
829    1  62.0      0      0  113572  80.0   B28      NaN   Mrs
```

[39]:
```
df_titanic.loc[df_titanic['Embarked'] == "C"]['Pclass'].mean()
```

[39]: 1.8518518518518519

[40]:
```
df_titanic['Embarked'].fillna('C', inplace = True)
```

[41]:
```
df['Embarked'] = df_titanic['Embarked']

df
```

[41]:
```
      Survived  Pclass  Sex   Title Embarked
0          0.0       3    0      Mr        S
1          1.0       1    1     Mrs        C
2          1.0       3    1    Miss        S
3          1.0       1    1     Mrs        S
```

```
4           0.0       3    0      Mr        S
…          …      …  …      …       …
1304        NaN       3    0      Mr        S
1305        NaN       1    1  Person        C
1306        NaN       3    0      Mr        S
1307        NaN       3    0      Mr        S
1308        NaN       3    0  Master        C

[1309 rows x 5 columns]
```

[ ]:

```

```

1.11 Tratar Title, Pclass, Embarked com Getdummies sem dropar first

[42]:

```python
pclass = pd.get_dummies(df['Pclass'], prefix = "Pclass")
title = pd.get_dummies(df['Title'], prefix = 'Title')
embarked = pd.get_dummies(df['Embarked'], prefix = 'Embarked')

df2 = pd.concat([df, pclass, title, embarked], axis = 1)
df2.drop(['Pclass', 'Title', 'Embarked'], axis=1, inplace=True)
```

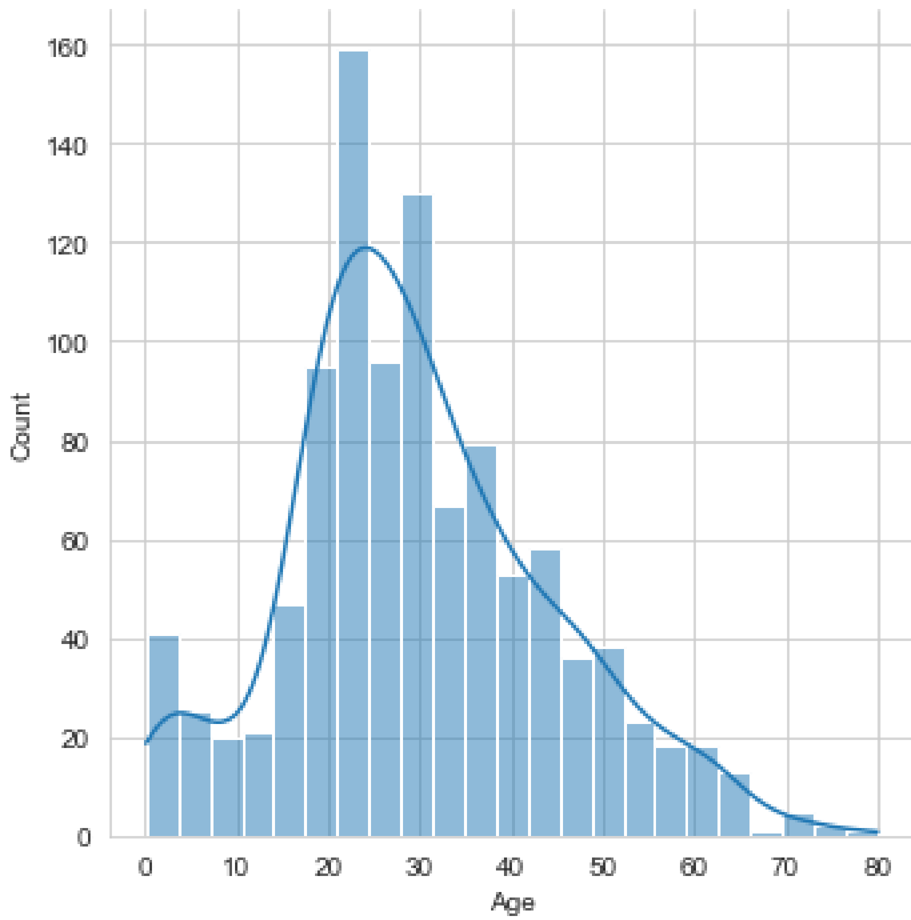
1.12 Tratando idade (parte 1)

[43]:

```python
df_info(df_titanic, 'Age', False)
```

```
Quantidade de valores únicos na Age:
98

Quais são os valores únicos na Age:
[22.   38.   26.   35.     nan 54.    2.   27.   14.    4.   58.   20.
 39.   55.   31.   34.   15.   28.    8.   19.   40.   66.   42.   21.
 18.    3.    7.   49.   29.   65.   28.5   5.   11.   45.   17.   32.
 16.   25.    0.83 30.   33.   23.   24.   46.   59.   71.   37.   47.
 14.5  70.5  32.5  12.    9.   36.5  51.   55.5  40.5  44.    1.   61.
 56.   50.   36.   45.5  20.5  62.   41.   52.   63.   23.5   0.92 43.
 60.   10.   64.   13.   48.    0.75 53.   57.   80.   70.   24.5   6.
  0.67 30.5   0.42 34.5  74.   22.5  18.5  67.   76.   26.5  60.5  11.5
  0.33  0.17 38.5 ]

Quantidade de valores nulos na Age:
263

Quantidade por opção na Age:
24.0    47
22.0    43
21.0    41
30.0    40
```

```
18.0    39
        ..
23.5     1
70.5     1
55.5     1
20.5     1
38.5     1
Name: Age, Length: 98, dtype: int64
```



[44]:
```
df2['Age'] = df_titanic['Age']
df2
```

[44]:

```
   Survived  Sex  Pclass_1  Pclass_2  Pclass_3  Title_Master  Title_Miss  \
0       0.0    0         0         0         1             0           0
1       1.0    1         1         0         0             0           0
2       1.0    1         0         0         1             0           1
3       1.0    1         1         0         0             0           0
4       0.0    0         0         0         1             0           0
```

```
…         …  …       …         …         …         …         …
1304         NaN     0         0         0         1             0             0
1305         NaN     1         1         0         0             0             0
1306         NaN     0         0         0         1             0             0
1307         NaN     0         0         0         1             0             0
1308         NaN     0         0         0         1             1             0

      Title_Mr  Title_Mrs  Title_Person  Embarked_C  Embarked_Q  Embarked_S  \
0            1          0             0           0           0           1
1            0          1             0           1           0           0
2            0          0             0           0           0           1
3            0          1             0           0           0           1
4            1          0             0           0           0           1
…         …         …           …          …          …          …
1304         1          0             0           0           0           1
1305         0          0             1           1           0           0
1306         1          0             0           0           0           1
1307         1          0             0           0           0           1
1308         0          0             0           1           0           0

       Age
0     22.0
1     38.0
2     26.0
3     35.0
4     35.0
…      …
1304   NaN
1305  39.0
1306  38.5
1307   NaN
1308   NaN

[1309 rows x 14 columns]
```

1.13 Encontrando melhor correlação para preencher valores vazios para idade

[45]:
```
df_titanic.corr()
```

[45]:
```
             PassengerId  Survived    Pclass       Sex       Age     SibSp  \
PassengerId     1.000000 -0.005007 -0.038354 -0.013406  0.028814 -0.055224
Survived       -0.005007  1.000000 -0.338481  0.543351 -0.077221 -0.035322
Pclass         -0.038354 -0.338481  1.000000 -0.124617 -0.408106  0.060832
Sex            -0.013406  0.543351 -0.124617  1.000000 -0.063645  0.109609
Age             0.028814 -0.077221 -0.408106 -0.063645  1.000000 -0.243699
SibSp          -0.055224 -0.035322  0.060832  0.109609 -0.243699  1.000000
Parch           0.008942  0.081629  0.018322  0.213125 -0.150917  0.373587
```

```
Fare           0.031428  0.257307 -0.558629  0.185523  0.178740  0.160238

                 Parch      Fare
PassengerId  0.008942  0.031428
Survived     0.081629  0.257307
Pclass       0.018322 -0.558629
Sex          0.213125  0.185523
Age         -0.150917  0.178740
SibSp        0.373587  0.160238
Parch        1.000000  0.221539
Fare         0.221539  1.000000
```

[46]:
```
#PCLASS POSSUI MAIOR MODULO DE CORRELAÇÃO PARA IDADE, IREMOS PREENCHER OS␣
↪VAZIOS BASEADOS
```

1.14 Encontrando media de idades baseados na classe e no titulo

[47]:
```
df2.loc[(df2['Pclass_1'] == 1) & (df2['Title_Master'] == 1)]['Age']
```

[47]:
```
305      0.92
445      4.00
802     11.00
955     13.00
1087     6.00
Name: Age, dtype: float64
```

[48]:
```
pclass1_master_mean_age = df2.loc[(df2['Pclass_1'] == 1) & (df2['Title_Master']␣
↪== 1)]['Age'].mean()
```

[49]:
```
pclass1_master_mean_age
```

[49]:
```
6.984
```

[50]:
```
pclass_1_miss_mean_age = df2.loc[(df2['Pclass_1'] == 1) & (df2['Title_Miss'] ==␣
↪1)]['Age'].mean()
```

[51]:
```
pclass_1_miss_mean_age
```

[51]:
```
30.338983050847457
```

[52]:
```
pclass_1_mr_mean_age = df2.loc[(df2['Pclass_1'] == 1) & (df2['Title_Mr'] ==␣
↪1)]['Age'].mean()
```

[53]:
```
pclass_1_mr_mean_age
```

[53]:
```
41.45075757575758
```

[54]:
```
pclass_1_person_mean_age = df2.loc[(df2['Pclass_1'] == 1) &
    ↪(df2['Title_Person'] == 1)]['Age'].mean()
```

[55]:
```
pclass_1_person_mean_age
```

[55]: 44.285714285714285

[56]:
```
pclass2_master_mean_age = df2.loc[(df2['Pclass_2'] == 1) & (df2['Title_Master']
    ↪== 1)]['Age'].mean()
```

[57]:
```
pclass2_master_mean_age
```

[57]: 2.7572727272727273

[58]:
```
pclass2_miss_mean_age = df2.loc[(df2['Pclass_2'] == 1) & (df2['Title_Miss'] ==
    ↪1)]['Age'].mean()
```

[59]:
```
pclass2_miss_mean_age
```

[59]: 20.717083333333333

[60]:
```
pclass2_mr_mean_age = df2.loc[(df2['Pclass_2'] == 1) & (df2['Title_Mr'] ==
    ↪1)]['Age'].mean()
```

[61]:
```
pclass2_mr_mean_age
```

[61]: 32.346715328467155

[62]:
```
pclass2_mrs_mean_age = df2.loc[(df2['Pclass_2'] == 1)& (df2['Title_Mrs'] ==
    ↪1)]['Age'].mean()
```

[63]:
```
pclass2_mrs_mean_age
```

[63]: 33.51851851851852

[64]:
```
pclass2_person_mean_age = df2.loc[(df2['Pclass_2'] == 1) & (df2['Title_Person']
    ↪== 1)]['Age'].mean()
```

[65]:
```
pclass2_person_mean_age
```

[65]: 39.54545454545455

[66]:
```
pclass3_master_mean_age = df2.loc[(df2['Pclass_3'] == 1) & (df2['Title_Master']
    ↪== 1)]['Age'].mean()
```

[67]:
```
pclass3_master_mean_age
```

[67]: 6.090000000000001

[68]:
```
pclass3_mr_mean_age = df2.loc[(df2['Pclass_3'] == 1) & (df2['Title_Mr'] ==␣
↪1)]['Age'].mean()
```

[69]:
```
pclass3_mr_mean_age
```

[69]: 28.318910256410255

[70]:
```
pclass3_mrs_mean_age = df2.loc[(df2['Pclass_3'] == 1) & (df2['Title_Mrs'] ==␣
↪1)]['Age'].mean()
```

[71]:
```
pclass3_mrs_mean_age
```

[71]: 32.326530612244895

[72]:
```
df2.loc[(df2['Pclass_3'] == 1) & (df2['Title_Person'] == 1)]
```

[72]:
```
     Survived  Sex  Pclass_1  Pclass_2  Pclass_3  Title_Master  Title_Miss  \
979       NaN    1         0         0         1             0           0

     Title_Mr  Title_Mrs  Title_Person  Embarked_C  Embarked_Q  Embarked_S  \
979         0          0             1           0           1           0

     Age
979  NaN
```

[73]:
```
pclass3_person_mean_age = df2.loc[(df2['Pclass_3'] == 1) & (df2['Title_Person']␣
↪== 1)]['Age'].mean()
```

[74]:
```
pclass3_person_mean_age
```

[74]: nan

[75]:
```
pclass3_mean = df2.loc[(df2['Pclass_3'] == 1)]['Age'].mean()
pclass3_mean
```

[75]: 24.81636726546906

[76]:
```
df2.loc[(df2['Pclass_1'] == 1) & (df2['Title_Mr'] == 1)]['Age'].isnull().sum()
```

[76]: 27

[77]:
```
pclasses = ['Pclass_1, Pclass_2, Pclass_3']
titles = ['Title_Master', 'Title_Mr', 'Title_Mrs', 'Title_Miss', 'Title_Person']
df2.loc[2, 'Age']
```

[77]: 26.0

[78]:
```
for i in df2.index:

    if pd.isnull(df2['Age'][i]):
#         for classe, titulo in [(classe,titulo) for classe in pclasses and␣
↪titulo in titles]:
        classe = df.loc[i, 'Pclass']
        titulo = df.loc[i, 'Title']
        mean_age = round(df2.loc[(df2[f'Pclass_{classe}'] == 1) &␣
↪(df2[f'Title_{titulo}'] == 1)]['Age'].mean(), 0)
        mean_age_pclass = round(df2.loc[df2[f'Pclass_{classe}'] == 1]['Age'].
↪mean(),0)
        if np.isnan(mean_age) == False:
            df2.loc[i, 'Age'] = mean_age
        else:
            df2.loc[i, 'Age'] = mean_age_pclass
df2['Age'].isnull().sum()
```

[78]: 0

1.15 Encontrando média de idade por Title

[79]:
```
df_titanic
```

[79]:
```
      PassengerId  Survived  Pclass  \
0               1       0.0       3
1               2       1.0       1
2               3       1.0       3
3               4       1.0       1
4               5       0.0       3
…             …         …       …
1304         1305       NaN       3
1305         1306       NaN       1
1306         1307       NaN       3
1307         1308       NaN       3
1308         1309       NaN       3

                                                   Name  Sex   Age  SibSp  \
0                               Braund, Mr. Owen Harris    0  22.0      1
1     Cumings, Mrs. John Bradley (Florence Briggs Th…    1  38.0      1
2                                Heikkinen, Miss. Laina    1  26.0      0
3          Futrelle, Mrs. Jacques Heath (Lily May Peel)    1  35.0      1
4                              Allen, Mr. William Henry    0  35.0      0
…                                                   …  …    …    …
1304                                 Spector, Mr. Woolf    0   NaN      0
1305                       Oliva y Ocana, Dona. Fermina    1  39.0      0
1306                       Saether, Mr. Simon Sivertsen    0  38.5      0
1307                                Ware, Mr. Frederick    0   NaN      0
```

```
1308                          Peter, Master. Michael J    0   NaN      1

      Parch              Ticket      Fare Cabin Embarked   Title
0         0           A/5 21171    7.2500   NaN        S      Mr
1         0            PC 17599   71.2833   C85        C     Mrs
2         0    STON/O2. 3101282    7.9250   NaN        S    Miss
3         0              113803   53.1000  C123        S     Mrs
4         0              373450    8.0500   NaN        S      Mr
…       …                   …        …     …       …       …
1304      0           A.5. 3236    8.0500   NaN        S      Mr
1305      0            PC 17758  108.9000  C105        C  Person
1306      0  SOTON/O.Q. 3101262    7.2500   NaN        S      Mr
1307      0              359309    8.0500   NaN        S      Mr
1308      1                2668   22.3583   NaN        C  Master

[1309 rows x 13 columns]
```

[80]:
```
df2
```

[80]:
```
      Survived  Sex  Pclass_1  Pclass_2  Pclass_3  Title_Master  Title_Miss  \
0          0.0    0         0         0         1             0           0
1          1.0    1         1         0         0             0           0
2          1.0    1         0         0         1             0           1
3          1.0    1         1         0         0             0           0
4          0.0    0         0         0         1             0           0
…          …   …        …         …         …            …          …
1304       NaN    0         0         0         1             0           0
1305       NaN    1         1         0         0             0           0
1306       NaN    0         0         0         1             0           0
1307       NaN    0         0         0         1             0           0
1308       NaN    0         0         0         1             1           0

      Title_Mr  Title_Mrs  Title_Person  Embarked_C  Embarked_Q  Embarked_S  \
0            1          0             0           0           0           1
1            0          1             0           1           0           0
2            0          0             0           0           0           1
3            0          1             0           0           0           1
4            1          0             0           0           0           1
…          …        …           …          …          …          …
1304         1          0             0           0           0           1
1305         0          0             1           1           0           0
1306         1          0             0           0           0           1
1307         1          0             0           0           0           1
1308         0          0             0           1           0           0

       Age
0     22.0
```

```
1      38.0
2      26.0
3      35.0
4      35.0
…      …
1304   28.0
1305   39.0
1306   38.5
1307   28.0
1308    6.0

[1309 rows x 14 columns]
```

[81]:
```python
df2.drop(['Pclass_1', 'Title_Master', 'Embarked_C'], axis = 1, inplace = True)
df2
```

[81]:
```
      Survived  Sex  Pclass_2  Pclass_3  Title_Miss  Title_Mr  Title_Mrs  \
0          0.0    0         0         1           0         1          0
1          1.0    1         0         0           0         0          1
2          1.0    1         0         1           1         0          0
3          1.0    1         0         0           0         0          1
4          0.0    0         0         1           0         1          0
…          …    …         …         …           …         …
1304       NaN    0         0         1           0         1          0
1305       NaN    1         0         0           0         0          0
1306       NaN    0         0         1           0         1          0
1307       NaN    0         0         1           0         1          0
1308       NaN    0         0         1           0         0          0

      Title_Person  Embarked_Q  Embarked_S   Age
0                0           0           1  22.0
1                0           0           0  38.0
2                0           0           1  26.0
3                0           0           1  35.0
4                0           0           1  35.0
…             …           …           …   …
1304             0           0           1  28.0
1305             1           0           0  39.0
1306             0           0           1  38.5
1307             0           0           1  28.0
1308             0           0           0   6.0

[1309 rows x 11 columns]
```

[82]:
```python
df_titanic.head(2)
```

[82]:
```
   PassengerId  Survived  Pclass  \
0            1       0.0       3
1            2       1.0       1

                                                Name  Sex   Age  SibSp  Parch  \
0                            Braund, Mr. Owen Harris    0  22.0      1      0
1  Cumings, Mrs. John Bradley (Florence Briggs Th…    1  38.0      1      0

      Ticket     Fare Cabin Embarked Title
0  A/5 21171   7.2500   NaN        S    Mr
1   PC 17599  71.2833   C85        C   Mrs
```

1.16 FamilySize

SibSp compreende a relação familiar como irmãos de sangue ou não + maridos/esposa Parch compreende a relação familiar como pai/padrastro, mae/madrasta, filhos de sangue ou nao

Poranto a o tamanho da familia é a pessoa + sibsp + parch

[83]:
```python
df2['FamilySize'] = df_titanic['SibSp'] + df_titanic['Parch'] + 1
df2
```

[83]:
```
      Survived  Sex  Pclass_2  Pclass_3  Title_Miss  Title_Mr  Title_Mrs  \
0          0.0    0         0         1           0         1          0
1          1.0    1         0         0           0         0          1
2          1.0    1         0         1           1         0          0
3          1.0    1         0         0           0         0          1
4          0.0    0         0         1           0         1          0
…           …   …         …         …         …          …         …
1304       NaN    0         0         1           0         1          0
1305       NaN    1         0         0           0         0          0
1306       NaN    0         0         1           0         1          0
1307       NaN    0         0         1           0         1          0
1308       NaN    0         0         1           0         0          0

      Title_Person  Embarked_Q  Embarked_S   Age  FamilySize
0                0           0           1  22.0           2
1                0           0           0  38.0           2
2                0           0           1  26.0           1
3                0           0           1  35.0           2
4                0           0           1  35.0           1
…             …          …          …    …          …
1304             0           0           1  28.0           1
1305             1           0           0  39.0           1
1306             0           0           1  38.5           1
1307             0           0           1  28.0           1
1308             0           0           0   6.0           3
```

```
[1309 rows x 12 columns]
```

1.17 Tratando Fare

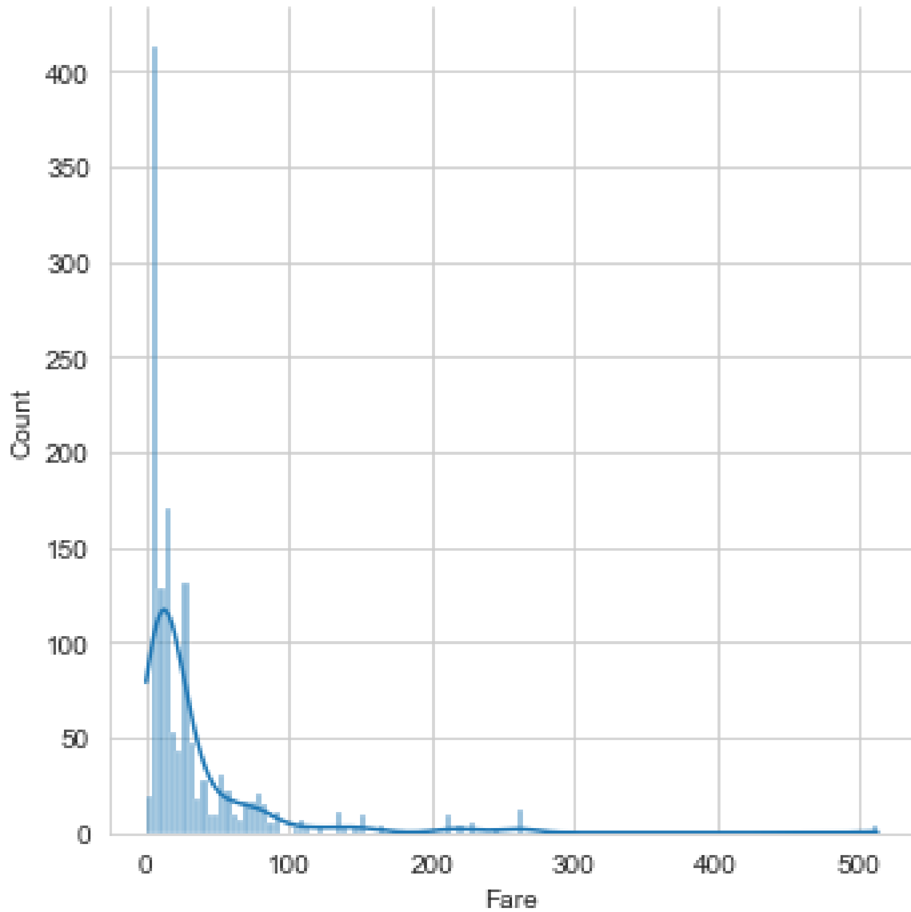
[84]:
```python
df_titanic.head(5)
```

[84]:
```
   PassengerId  Survived  Pclass  \
0            1       0.0       3
1            2       1.0       1
2            3       1.0       3
3            4       1.0       1
4            5       0.0       3

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

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

[85]:
```python
df_info(df_titanic, 'Fare', False)
```

```
Quantidade de valores únicos na Fare:
281

Quais são os valores únicos na Fare:
[  7.25     71.2833   7.925    53.1       8.05      8.4583   51.8625   21.075
  11.1333  30.0708  16.7      26.55     31.275     7.8542   16.       29.125
  13.      18.       7.225    26.        8.0292   35.5      31.3875  263.
   7.8792   7.8958  27.7208 146.5208    7.75     10.5      82.1708   52.
   7.2292  11.2417   9.475    21.      41.5792   15.5      21.6792   17.8
  39.6875   7.8     76.7292  61.9792   27.75     46.9      80.       83.475
  27.9     15.2458   8.1583    8.6625  73.5     14.4542   56.4958    7.65
  29.      12.475    9.        9.5      7.7875   47.1     15.85     34.375
  61.175   20.575   34.6542  63.3583  23.       77.2875    8.6542    7.775
  24.15     9.825   14.4583 247.5208    7.1417  22.3583    6.975     7.05
  14.5     15.0458  26.2833   9.2167  79.2       6.75     11.5      36.75
   7.7958  12.525   66.6       7.3125  61.3792    7.7333  69.55     16.1
  15.75    20.525   55.       25.925   33.5      30.6958  25.4667   28.7125
   0.      15.05    39.       22.025   50.        8.4042    6.4958  10.4625
  18.7875  31.     113.275    27.      76.2917  90.        9.35     13.5
```

```
   7.55     26.25     12.275     7.125    52.5542  20.2125  86.5     512.3292
  79.65   153.4625 135.6333  19.5      29.7      77.9583  20.25     78.85
  91.0792  12.875    8.85   151.55     30.5      23.25    12.35   110.8833
 108.9     24.      56.9292  83.1583 262.375   14.     164.8667 134.5
   6.2375  57.9792  28.5    133.65     15.9      9.225   35.      75.25
  69.3     55.4417 211.5      4.0125 227.525   15.7417   7.7292  12.
 120.      12.65    18.75     6.8583  32.5      7.875   14.4     55.9
   8.1125  81.8583  19.2583  19.9667  89.1042  38.5      7.725   13.7917
   9.8375   7.0458   7.5208  12.2875   9.5875  49.5042  78.2667  15.1
   7.6292  22.525   26.2875  59.4      7.4958  34.0208  93.5    221.7792
 106.425   49.5     71.      13.8625   7.8292  39.6     17.4     51.4792
  26.3875  30.      40.125    8.7125  15.      33.      42.4     15.55
  65.      32.3208   7.0542   8.4333  25.5875   9.8417   8.1375  10.1708
 211.3375  57.      13.4167   7.7417   9.4833   7.7375   8.3625  23.45
  25.9292   8.6833   8.5167   7.8875  37.0042   6.45     6.95     8.3
   6.4375  39.4     14.1083  13.8583  50.4958   5.       9.8458  10.5167
   7.       9.6875  82.2667   3.1708  31.6833  31.5     57.75     7.85
  60.      15.0333  15.5792  28.5375  25.7     10.7083  13.9      7.8208
   7.7792  31.6792   7.2833  75.2417      nan  12.1833  13.775    8.9625
  25.7417  42.5     27.4458 136.7792   9.325   12.7375  45.5      7.575
   7.5792   7.7208]

Quantidade de valores nulos na Fare:
1

Quantidade por opção na Fare:
8.0500     60
13.0000    59
7.7500     55
26.0000    50
7.8958     49
           ..
7.7417      1
8.1583      1
8.4583      1
7.8000      1
7.7208      1
Name: Fare, Length: 281, dtype: int64
```



[86]:
```
df_titanic.loc[df_titanic['Fare'].isnull()]
```

[86]:
```
      PassengerId  Survived  Pclass                Name  Sex   Age  SibSp  \
1043         1044       NaN       3  Storey, Mr. Thomas    0  60.5      0

      Parch Ticket  Fare Cabin Embarked Title
1043      0   3701   NaN   NaN        S    Mr
```

[87]:
```
fare_mean_class3 = df_titanic.loc[(df_titanic['Pclass']  == 3)]['Fare'].mean()
fare_mean_class3
```

[87]: 13.302888700564969

[88]:
```
for i in df_titanic.index:
    if pd.isna(df_titanic.loc[i, 'Fare']):
        df_titanic.loc[i, 'Fare'] = fare_mean_class3

df_titanic['Fare'].isnull().sum()
```

[88]: 0

[89]:
```
df2['Fare'] = df_titanic['Fare']
df2['Fare'].isnull().sum()
```

[89]: 0

1.18 Separando as bases

[90]:
```
train = df2[:train_index].copy()
train
```

[90]:
```
     Survived  Sex  Pclass_2  Pclass_3  Title_Miss  Title_Mr  Title_Mrs  \
0         0.0    0         0         1           0         1          0
1         1.0    1         0         0           0         0          1
2         1.0    1         0         1           1         0          0
3         1.0    1         0         0           0         0          1
4         0.0    0         0         1           0         1          0
..        …   …         …         …           …         …          …
886       0.0    0         1         0           0         0          0
887       1.0    1         0         0           1         0          0
888       0.0    1         0         1           1         0          0
889       1.0    0         0         0           0         1          0
890       0.0    0         0         1           0         1          0

     Title_Person  Embarked_Q  Embarked_S   Age  FamilySize     Fare
0               0           0           1  22.0           2   7.2500
1               0           0           0  38.0           2  71.2833
2               0           0           1  26.0           1   7.9250
3               0           0           1  35.0           2  53.1000
4               0           0           1  35.0           1   8.0500
..            …           …           …   …          …        …
886             1           0           1  27.0           1  13.0000
887             0           0           1  19.0           1  30.0000
888             0           0           1  17.0           4  23.4500
889             0           0           0  26.0           1  30.0000
890             0           1           0  32.0           1   7.7500

[891 rows x 13 columns]
```

[91]:
```
test = df2[test_index:].copy()
test
```

[91]:
```
     Survived  Sex  Pclass_2  Pclass_3  Title_Miss  Title_Mr  Title_Mrs  \
891       NaN    0         0         1           0         1          0
892       NaN    1         0         1           0         0          1
893       NaN    0         1         0           0         1          0
```

```
894        NaN    0         0         1           0         1          0
895        NaN    1         0         1           0         0          1
…        …  …       …       …         …       …        …
1304       NaN    0         0         1           0         1          0
1305       NaN    1         0         0           0         0          0
1306       NaN    0         0         1           0         1          0
1307       NaN    0         0         1           0         1          0
1308       NaN    0         0         1           0         0          0

      Title_Person  Embarked_Q  Embarked_S   Age  FamilySize      Fare
891              0           1           0  34.5           1    7.8292
892              0           0           1  47.0           2    7.0000
893              0           1           0  62.0           1    9.6875
894              0           0           1  27.0           1    8.6625
895              0           0           1  22.0           3   12.2875
…              …         …         …   …       …       …
1304             0           0           1  28.0           1    8.0500
1305             1           0           0  39.0           1  108.9000
1306             0           0           1  38.5           1    7.2500
1307             0           0           1  28.0           1    8.0500
1308             0           0           0   6.0           3   22.3583

[418 rows x 13 columns]
```

[92]:
```python
train['Survived'] = train['Survived'].astype(int)
train
```

[92]:
```
     Survived  Sex  Pclass_2  Pclass_3  Title_Miss  Title_Mr  Title_Mrs  \
0           0    0         0         1           0         1          0
1           1    1         0         0           0         0          1
2           1    1         0         1           1         0          0
3           1    1         0         0           0         0          1
4           0    0         0         1           0         1          0
..        …  …       …       …         …       …        …
886         0    0         1         0           0         0          0
887         1    1         0         0           1         0          0
888         0    1         0         1           1         0          0
889         1    0         0         0           0         1          0
890         0    0         0         1           0         1          0

     Title_Person  Embarked_Q  Embarked_S   Age  FamilySize     Fare
0               0           0           1  22.0           2   7.2500
1               0           0           0  38.0           2  71.2833
2               0           0           1  26.0           1   7.9250
3               0           0           1  35.0           2  53.1000
4               0           0           1  35.0           1   8.0500
..            …         …         …   …       …      …
```

```
886            1          0          1  27.0          1  13.0000
887            0          0          1  19.0          1  30.0000
888            0          0          1  17.0          4  23.4500
889            0          0          0  26.0          1  30.0000
890            0          1          0  32.0          1   7.7500

[891 rows x 13 columns]
```

1.19 Definindo as variaveis X, y que irão no modelo

[93]:
```python
X = train.drop('Survived', axis = 1)
y = train['Survived']
```

[94]:
```python
X_test = test.drop('Survived', axis = 1)
```

[108]:
```python
def acuracia_algoritmo(algoritmo, X_train, y_train, vc):
    modelo = algoritmo.fit(X_train, y_train)
    acuracia = round(modelo.score(X_train, y_train) *100, 2)

    train_pred = model_selection.cross_val_predict(algoritmo, X_train, y_train,␣
 ↪cv = vc, n_jobs = -1)
    acuracia_vc = round(metrics.accuracy_score(y_train, train_pred) *100, 2)

    print(f"Acurácia: {acuracia}")
    print(f"Acurácia Validação Cruzada: {acuracia_vc}")
```

[ ]:
```python

```

1.20 Testando todos os Classificadores do SkLearn

1.21 Random Forest

[109]:
```python
acuracia_algoritmo(RandomForestClassifier(), X, y, 10)
```

```
Acurácia: 98.32
Acurácia Validação Cruzada: 81.37
```

1.22 Logistic Regression

[110]:
```python
acuracia_algoritmo(LogisticRegression(max_iter=1000), X, y, 10)
```

```
Acurácia: 82.94
Acurácia Validação Cruzada: 82.83
```

1.23 Gausian Naives Bayes

[111]:
```
acuracia_algoritmo(GaussianNB(), X, y, 10)
```

```
Acurácia: 78.0
Acurácia Validação Cruzada: 77.89
```

1.24 Linear Support Vector Machines (SVC)

[112]:
```
acuracia_algoritmo(LinearSVC(dual = False), X, y, 10)
```

```
Acurácia: 83.28
Acurácia Validação Cruzada: 82.94
```

1.25 K-nearest Neighbours

[114]:
```
acuracia_algoritmo(KNeighborsClassifier(), X, y, 10)
```

```
Acurácia: 81.59
Acurácia Validação Cruzada: 71.94
```

1.26 Stochastic Gradient Descent

[115]:
```
acuracia_algoritmo(SGDClassifier(), X, y, 10)
```

```
Acurácia: 73.18
Acurácia Validação Cruzada: 73.4
```

1.27 Decision Tree Classifier

[116]:
```
acuracia_algoritmo(DecisionTreeClassifier(), X, y, 10)
```

```
Acurácia: 98.32
Acurácia Validação Cruzada: 79.69
```

1.28 Gradient Boost Trees

[117]:
```
acuracia_algoritmo(GradientBoostingClassifier(), X, y, 10)
```

```
Acurácia: 90.24
Acurácia Validação Cruzada: 83.28
```

1.29 Treinando automaticamente os melhores parametros do melhor classificador no GridSearch para encontrar a melhor perfomance

[131]:
```
params = dict(
    max_depth = [n for n in range(1, 5)],
    min_samples_split = [n for n in range(2, 6)],
    min_samples_leaf = [n for n in range(2, 6)],
    n_estimators = [n for n in range(10, 50, 10)],
```

```
)
```

[132]:
```
gbc = GradientBoostingClassifier ()
```

1.30 Melhor classificador GRADIENT BOOSTING - Utilizando GrideSearch para melhorar a perfomace

[133]:
```
gbc_cv = GridSearchCV(estimator = gbc, param_grid = params, cv = 10)
```

[134]:
```
gbc_cv.fit(X, y)
```

[134]:
```
GridSearchCV(cv=10, estimator=GradientBoostingClassifier(),
             param_grid={'max_depth': [1, 2, 3, 4],
                         'min_samples_leaf': [2, 3, 4, 5],
                         'min_samples_split': [2, 3, 4, 5],
                         'n_estimators': [10, 20, 30, 40]})
```

[135]:
```
print(f"Melhor pontuação: {gbc_cv.best_score_}")
print(f"Melhores parâmetros: {gbc_cv.best_estimator_}")
```

```
Melhor pontuação: 0.8440324594257179
Melhores parâmetros: GradientBoostingClassifier(max_depth=4, min_samples_leaf=2,
min_samples_split=5,
                           n_estimators=30)
```

[136]:
```
gradientBoostingClassifier_pred = gbc_cv.predict(X_test)
```

[138]:
```
kaggle = pd.DataFrame({'PassengerId': passengerID, 'Survived':␣
 ↪gradientBoostingClassifier_pred})
```

[139]:
```
kaggle
```

[139]:
```
     PassengerId  Survived
0            892         0
1            893         0
2            894         0
3            895         0
4            896         1
..           …        …
413         1305         0
414         1306         1
415         1307         0
416         1308         0
417         1309         1

[418 rows x 2 columns]
```

[140]:
```
kaggle.to_csv('./titanic_gradient_boosting_pred.csv', index=False)
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: