ADS Assignment - 2

May 28, 2023

1 Titanic ship case study

Problem Description: On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. Translated 32% survival rate.

1. One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew.
2. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper- class.

The problem associated with the Titanic dataset is to predict whether a passenger survived the disaster or not. The dataset contains various features such as passenger class, age, gender, cabin, fare, and whether the passenger had any siblings or spouses on board. These features can be used to build a predictive model to determine the likelihood of a passenger surviving the disaster. The dataset offers opportunities for feature engineering, data visualization, and model selection, making it a valuable resource for developing and testing data analysis and machine learning skills.

1.1 Import necessary libraries

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

1. Download the dataset:

[2]:
```
# titanic.csv dataset downloaded and placed in the working directory
```

2. Load the dataset.

[3]:
```
data = pd.read_csv("titanic.csv")
```

[4]:
```
data.head()
```

[4]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	\
0	0	3	male	22.0	1	0	7.2500	S	Third	
1	1	1	female	38.0	1	0	71.2833	C	First	
2	1	3	female	26.0	0	0	7.9250	S	Third	
3	1	1	female	35.0	1	0	53.1000	S	First	

```
4         0       3    male  35.0      0      0   8.0500        S  Third

     who  adult_male deck  embark_town alive  alone
0    man        True  NaN  Southampton    no  False
1  woman       False    C    Cherbourg   yes  False
2  woman       False  NaN  Southampton   yes   True
3  woman       False    C  Southampton   yes  False
4    man        True  NaN  Southampton    no   True
```

[5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
 8   class        891 non-null    object
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    object
 12  embark_town  889 non-null    object
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
dtypes: bool(2), float64(2), int64(4), object(7)
memory usage: 92.4+ KB
```

3. Perform Below Visualizations.
   Univariate Analysis
   Bi - Variate Analysis
   Multi - Variate Analysis

1.2 Univariate Analysis

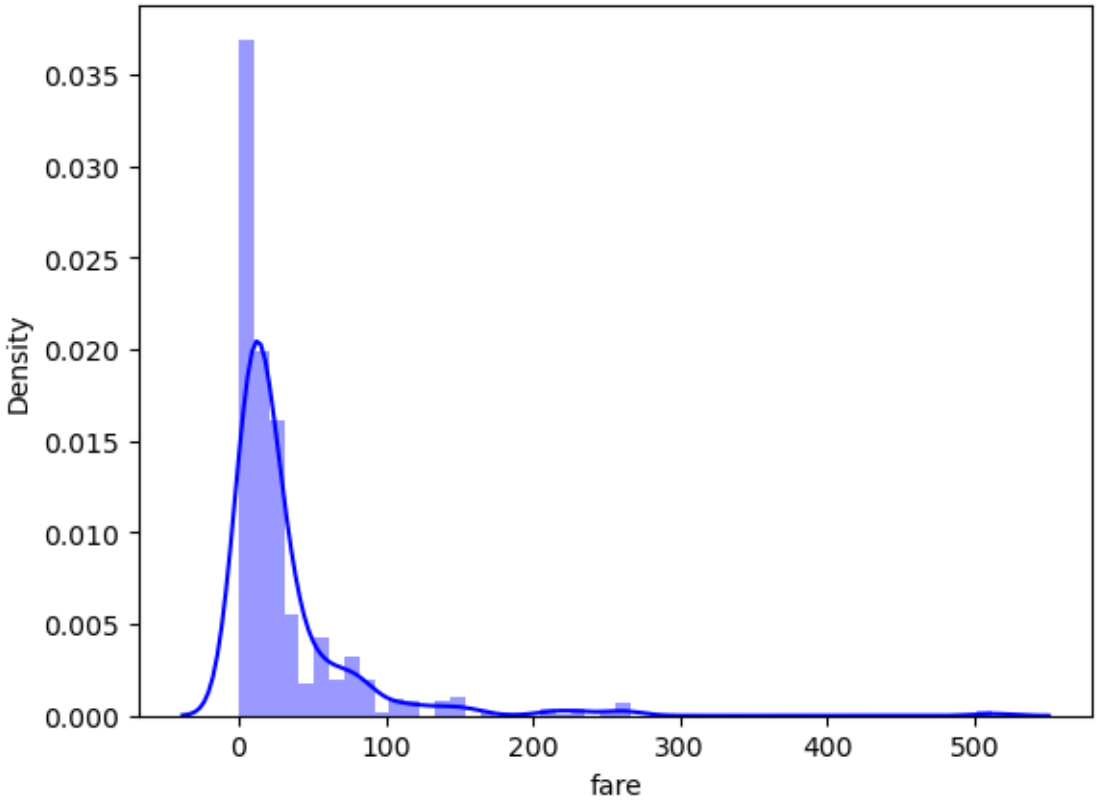
1.2.1 Distribution plot

[6]:
```
sns.distplot(data['fare'], color = 'b')
```

```
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/1361863019.py:1:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
```

```
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['fare'], color = 'b')
```

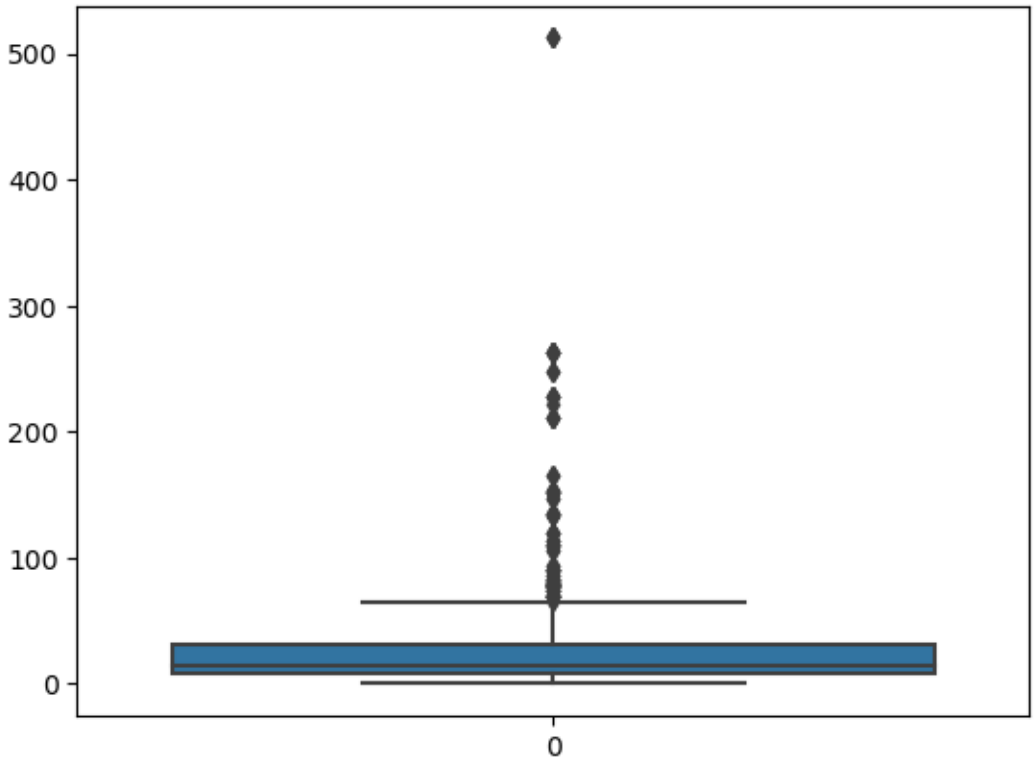
[6]: <Axes: xlabel='fare', ylabel='Density'>



1.2.2 Box plot

[7]:
```
sns.boxplot(data['fare'])
```

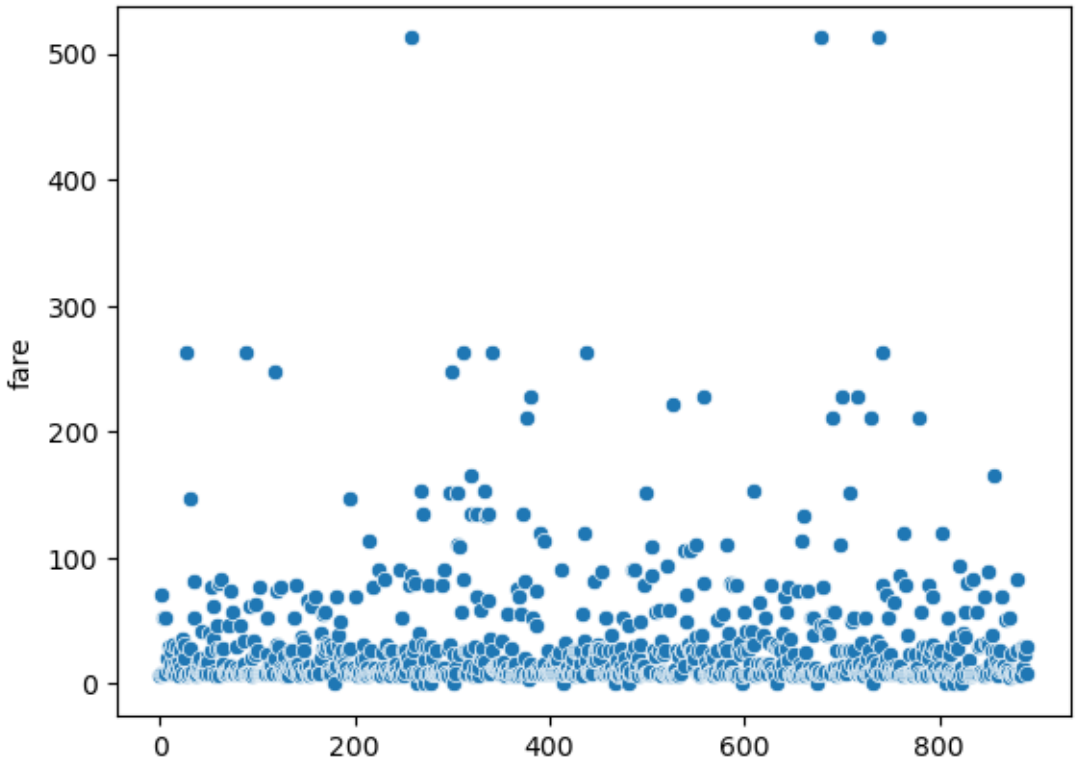
[7]: <Axes: >



1.2.3 Scatter plot

[8]:
```
sns.scatterplot(data['fare'])
```

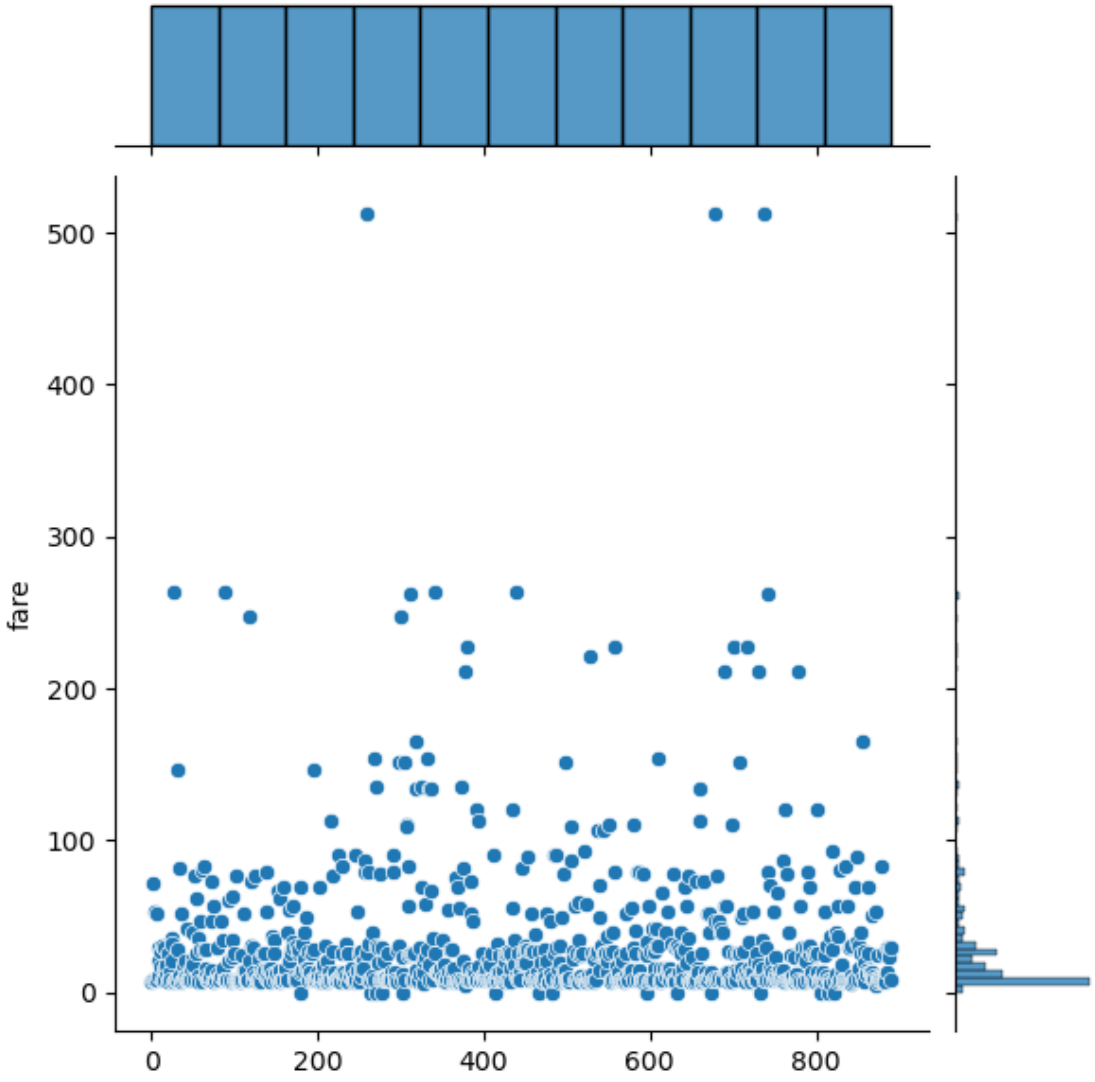
[8]:
```
<Axes: ylabel='fare'>
```



1.2.4 Joint plot

[9]:
```
sns.jointplot(data['fare'])
```

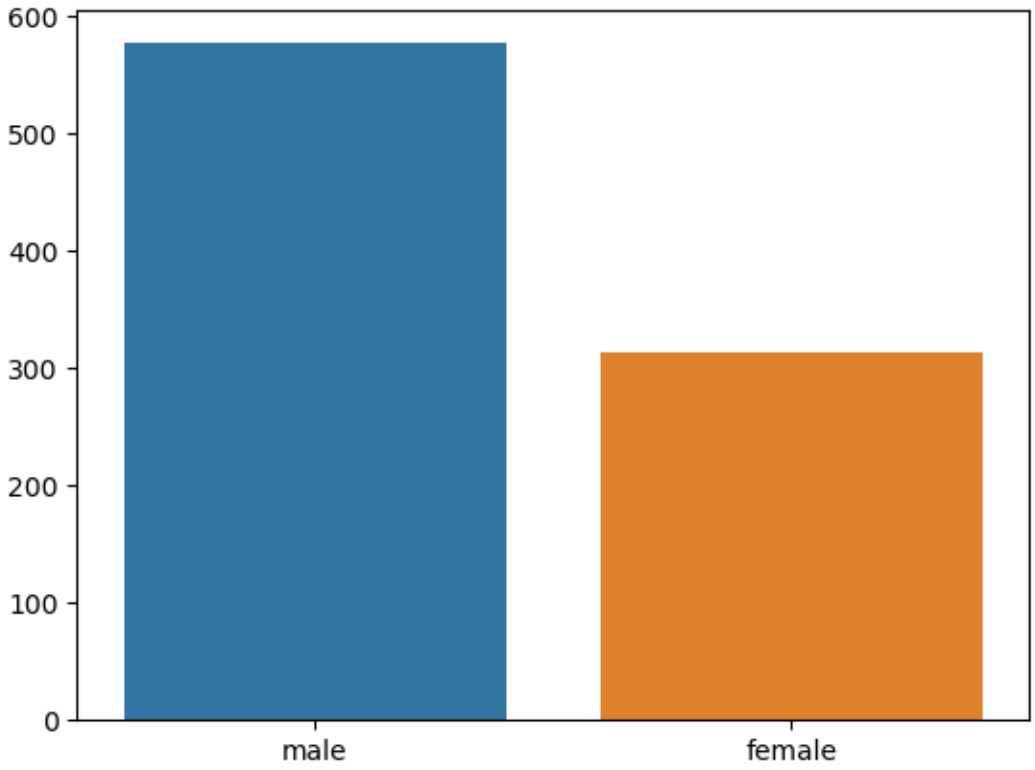
[9]:
```
<seaborn.axisgrid.JointGrid at 0x11cce1540>
```



1.2.5 Bar plot

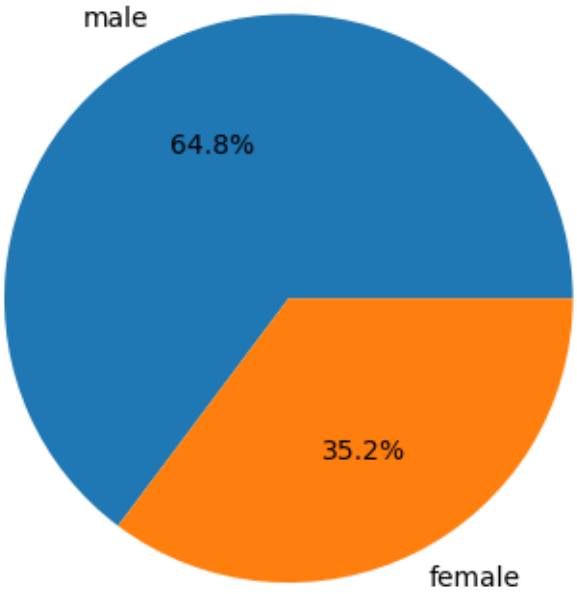
[10]:
```
x = data.sex.value_counts()
sns.barplot(x=x.index, y=x.values)
```

[10]: <Axes: >



1.2.6 Pie plot

[11]:
```
x = data['sex'].value_counts()
plt.pie(x.values,
        labels=x.index,
        autopct='%1.1f%%')
plt.show()
```

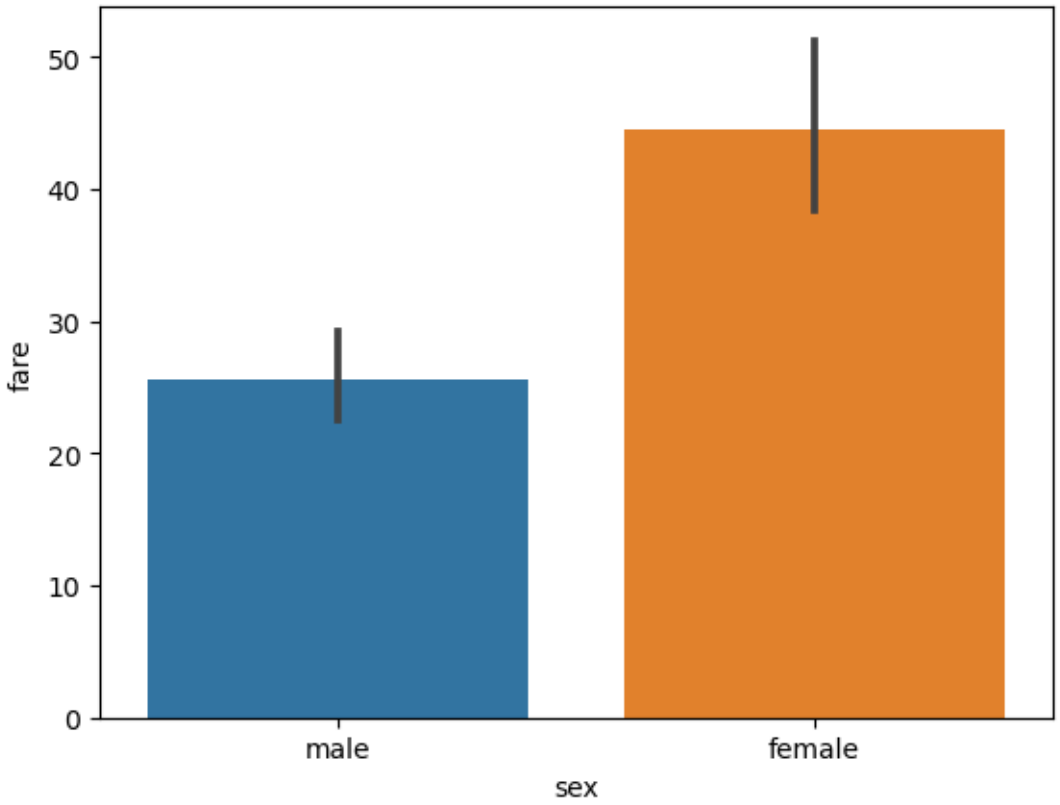


1.3 Bivariate analysis

1.3.1 Bar plot

[12]:
```
sns.barplot(x=data.sex, y=data.fare)
```

[12]:
```
<Axes: xlabel='sex', ylabel='fare'>
```

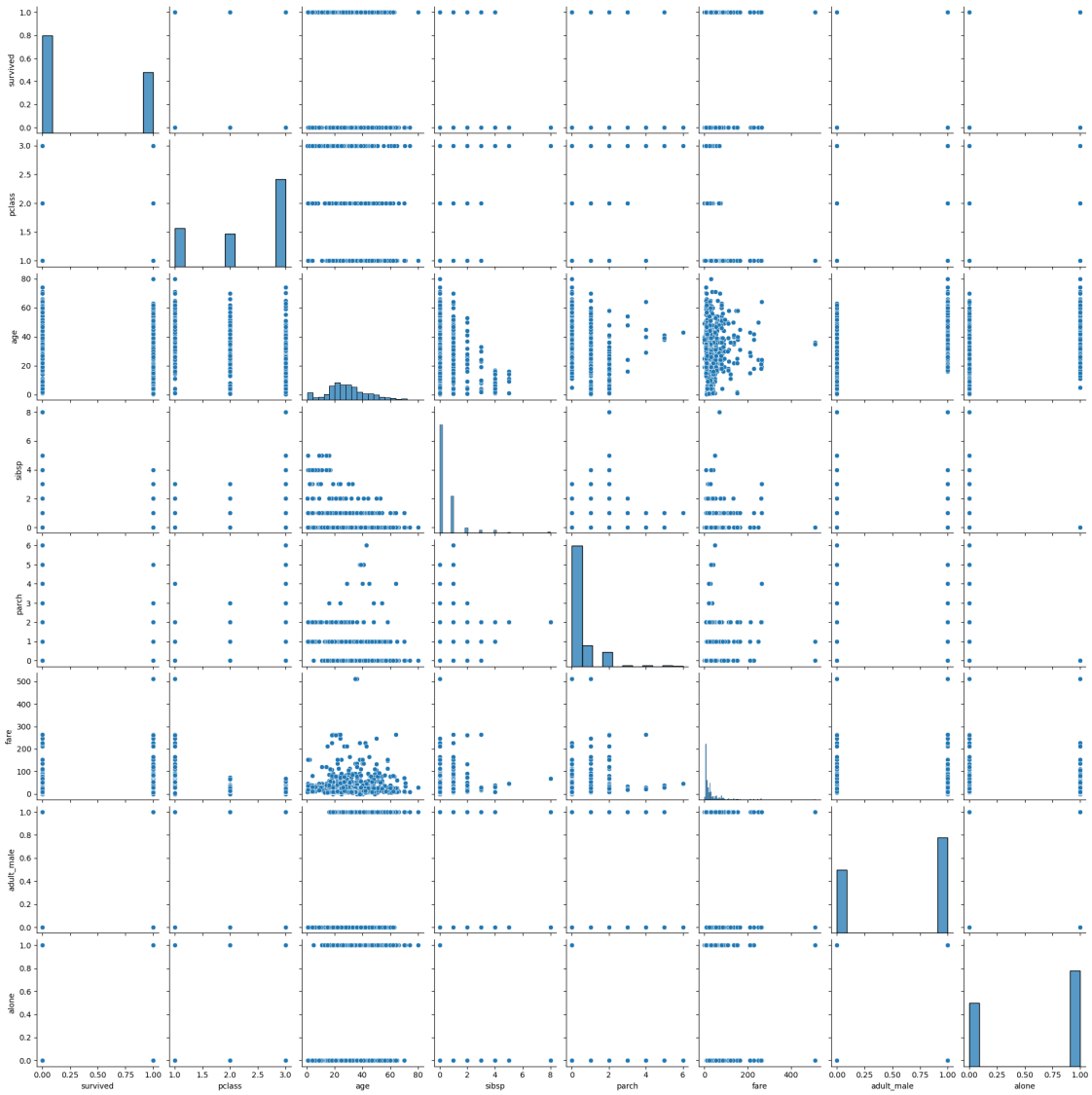


1.3.2 Pair plot

[13]:
```
sns.pairplot(data)
```

```
<__array_function__ internals>:180: RuntimeWarning: Converting input from bool
to <class 'numpy.uint8'> for compatibility.
<__array_function__ internals>:180: RuntimeWarning: Converting input from bool
to <class 'numpy.uint8'> for compatibility.
```

[13]:
```
<seaborn.axisgrid.PairGrid at 0x11d19a440>
```

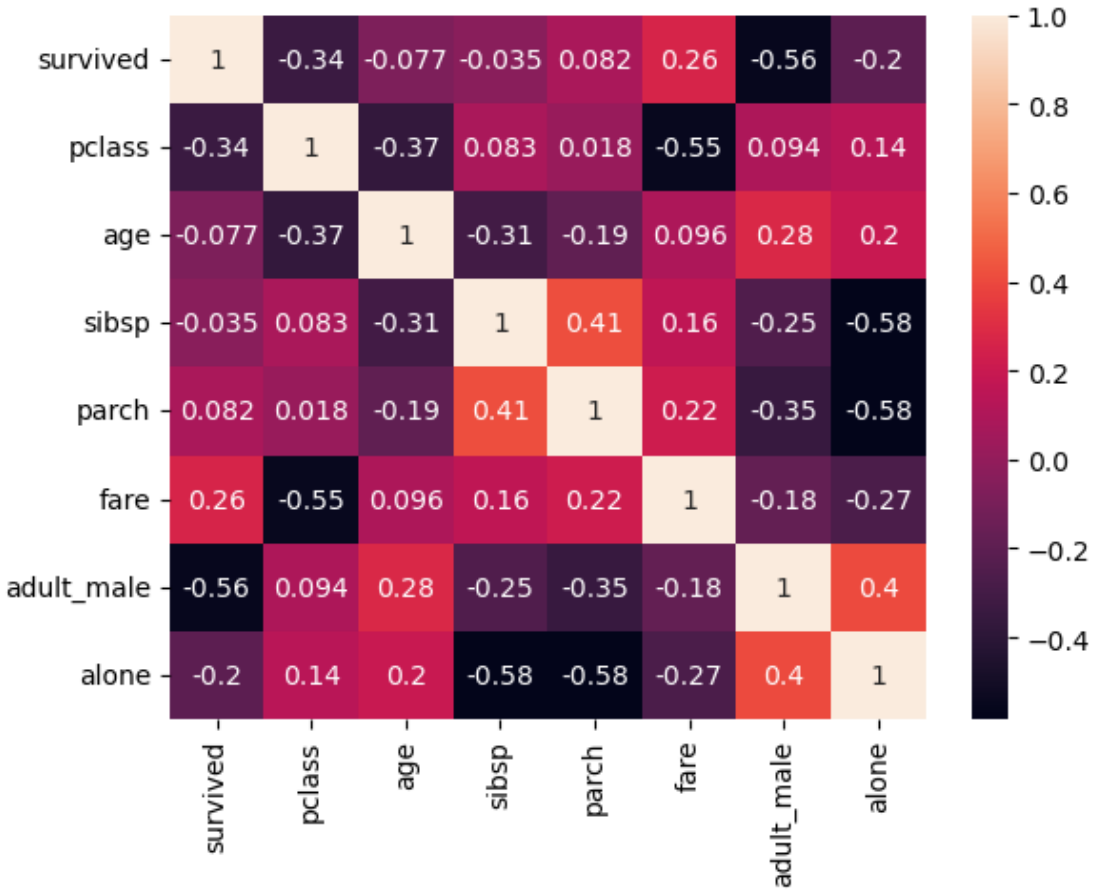


1.4 Multivariate Analysis

[14]:
```
sns.heatmap(data.corr(), annot=True)
```

```
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/1119197534.py:1:
FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only valid
columns or specify the value of numeric_only to silence this warning.
  sns.heatmap(data.corr(), annot=True)
```

[14]: <Axes: >



4. Perform descriptive statistics on the dataset.

1.5 Measure of central tendency - Mean, Median and Mode

[15]:
```
data.mean()
```

```
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/531903386.py:1:
FutureWarning: The default value of numeric_only in DataFrame.mean is
deprecated. In a future version, it will default to False. In addition,
specifying 'numeric_only=None' is deprecated. Select only valid columns or
specify the value of numeric_only to silence this warning.
  data.mean()
```

[15]:
```
survived      0.383838
pclass        2.308642
age          29.699118
sibsp         0.523008
parch         0.381594
fare         32.204208
```

```
adult_male     0.602694
alone          0.602694
dtype: float64
```

[16]:
```
data.median()
```

```
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/4184645713.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median is
deprecated. In a future version, it will default to False. In addition,
specifying 'numeric_only=None' is deprecated. Select only valid columns or
specify the value of numeric_only to silence this warning.
  data.median()
```

[16]:
```
survived       0.0000
pclass         3.0000
age           28.0000
sibsp          0.0000
parch          0.0000
fare          14.4542
adult_male     1.0000
alone          1.0000
dtype: float64
```

[17]:
```
data.mode()
```

[17]:
```
   survived  pclass   sex   age  sibsp  parch  fare embarked  class  who  \
0         0       3  male  24.0      0      0  8.05        S  Third  man

   adult_male deck  embark_town alive  alone
0        True    C  Southampton    no   True
```

1.6 Measure of variability

1.6.1 Kurtosis

[18]:
```
data.kurt()
```

```
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/2907027414.py:1:
FutureWarning: The default value of numeric_only in DataFrame.kurt is
deprecated. In a future version, it will default to False. In addition,
specifying 'numeric_only=None' is deprecated. Select only valid columns or
specify the value of numeric_only to silence this warning.
  data.kurt()
```

[18]:
```
survived      -1.775005
pclass        -1.280015
age            0.178274
sibsp         17.880420
parch          9.778125
```

```
fare          33.398141
adult_male    -1.827345
alone         -1.827345
dtype: float64
```

1.6.2 Range

[19]:
```
data.max()
```

```
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/2904433368.py:1:
FutureWarning: The default value of numeric_only in DataFrame.max is deprecated.
In a future version, it will default to False. In addition, specifying
'numeric_only=None' is deprecated. Select only valid columns or specify the
value of numeric_only to silence this warning.
  data.max()
```

[19]:
```
survived            1
pclass              3
sex              male
age              80.0
sibsp               8
parch               6
fare          512.3292
class            Third
who              woman
adult_male        True
alive              yes
alone             True
dtype: object
```

[20]:
```
data.min()
```

```
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/927168777.py:1:
FutureWarning: The default value of numeric_only in DataFrame.min is deprecated.
In a future version, it will default to False. In addition, specifying
'numeric_only=None' is deprecated. Select only valid columns or specify the
value of numeric_only to silence this warning.
  data.min()
```

[20]:
```
survived          0
pclass            1
sex          female
age            0.42
sibsp             0
parch             0
fare            0.0
class         First
who           child
```

```
adult_male     False
alive             no
alone          False
dtype: object
```

[21]:
```
Range = data.max()[['age', 'fare']] - data.min()[['age', 'fare']]
print(Range)
```

```
age        79.58
fare    512.3292
dtype: object
```

```
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/1058199015.py:1:
FutureWarning: The default value of numeric_only in DataFrame.max is deprecated.
In a future version, it will default to False. In addition, specifying
'numeric_only=None' is deprecated. Select only valid columns or specify the
value of numeric_only to silence this warning.
  Range = data.max()[['age', 'fare']] - data.min()[['age', 'fare']]
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/1058199015.py:1:
FutureWarning: The default value of numeric_only in DataFrame.min is deprecated.
In a future version, it will default to False. In addition, specifying
'numeric_only=None' is deprecated. Select only valid columns or specify the
value of numeric_only to silence this warning.
  Range = data.max()[['age', 'fare']] - data.min()[['age', 'fare']]
```

1.6.3 Skewness

[22]:
```
data.skew()
```

```
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/1188251951.py:1:
FutureWarning: The default value of numeric_only in DataFrame.skew is
deprecated. In a future version, it will default to False. In addition,
specifying 'numeric_only=None' is deprecated. Select only valid columns or
specify the value of numeric_only to silence this warning.
  data.skew()
```

[22]:
```
survived      0.478523
pclass       -0.630548
age           0.389108
sibsp         3.695352
parch         2.749117
fare          4.787317
adult_male   -0.420431
alone        -0.420431
dtype: float64
```

1.6.4 Interquartile range

[23]:
```
quantiles = data[['age', 'fare']].quantile(q=[0.75, 0.25])
quantiles
```

[23]:
```
          age     fare
0.75  38.000  31.0000
0.25  20.125   7.9104
```

[24]:
```
#Q3
quantiles.iloc[0]
```

[24]:
```
age     38.0
fare    31.0
Name: 0.75, dtype: float64
```

[25]:
```
#Q1
quantiles.iloc[1]
```

[25]:
```
age     20.1250
fare     7.9104
Name: 0.25, dtype: float64
```

[26]:
```
IQR = quantiles.iloc[0]-quantiles.iloc[1]
IQR
```

[26]:
```
age     17.8750
fare    23.0896
dtype: float64
```

1.6.5 Upper extreme

Q3 + 1.5*IQR

[27]:
```
quantiles.iloc[0] + (1.5*IQR)
```

[27]:
```
age     64.8125
fare    65.6344
dtype: float64
```

1.6.6 Lower extreme

Q1 - 1.5*IQR

[28]:
```
quantiles.iloc[1] - (1.5*IQR)
```

[28]:
```
age     -6.6875
fare   -26.7240
dtype: float64
```

1.6.7 Standard deviation

[29]:
```
data.std()
```

```
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/2723740006.py:1:
FutureWarning: The default value of numeric_only in DataFrame.std is deprecated.
In a future version, it will default to False. In addition, specifying
'numeric_only=None' is deprecated. Select only valid columns or specify the
value of numeric_only to silence this warning.
  data.std()
```

[29]:
```
survived       0.486592
pclass         0.836071
age           14.526497
sibsp          1.102743
parch          0.806057
fare          49.693429
adult_male     0.489615
alone          0.489615
dtype: float64
```

1.6.8 Variance

[30]:
```
data.var()
```

```
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/445316826.py:1:
FutureWarning: The default value of numeric_only in DataFrame.var is deprecated.
In a future version, it will default to False. In addition, specifying
'numeric_only=None' is deprecated. Select only valid columns or specify the
value of numeric_only to silence this warning.
  data.var()
```

[30]:
```
survived         0.236772
pclass           0.699015
age            211.019125
sibsp            1.216043
parch            0.649728
fare          2469.436846
adult_male       0.239723
alone            0.239723
dtype: float64
```

[31]:
```
data.describe()
```

[31]:

	survived	pclass	age	sibsp	parch	fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000

```
25%      0.000000    2.000000   20.125000    0.000000    0.000000    7.910400
50%      0.000000    3.000000   28.000000    0.000000    0.000000   14.454200
75%      1.000000    3.000000   38.000000    1.000000    0.000000   31.000000
max      1.000000    3.000000   80.000000    8.000000    6.000000  512.329200
```

5. Handle the Missing values.

[32]:
```
data.isnull().sum()
```

[32]:
```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

1.7 Handling missing value

[33]:
```
data['age'].fillna(data['age'].mean(), inplace=True)
```

[34]:
```
data['embarked'].fillna(data['embarked'].mode()[0], inplace=True)
```

[35]:
```
data['deck'].fillna(data['deck'].mode()[0], inplace=True)
```

[36]:
```
data['embark_town'].fillna(data['embark_town'].mode()[0], inplace=True)
```

[37]:
```
data.isnull().sum()
```

[37]:
```
survived       0
pclass         0
sex            0
age            0
sibsp          0
parch          0
fare           0
embarked       0
class          0
who            0
```

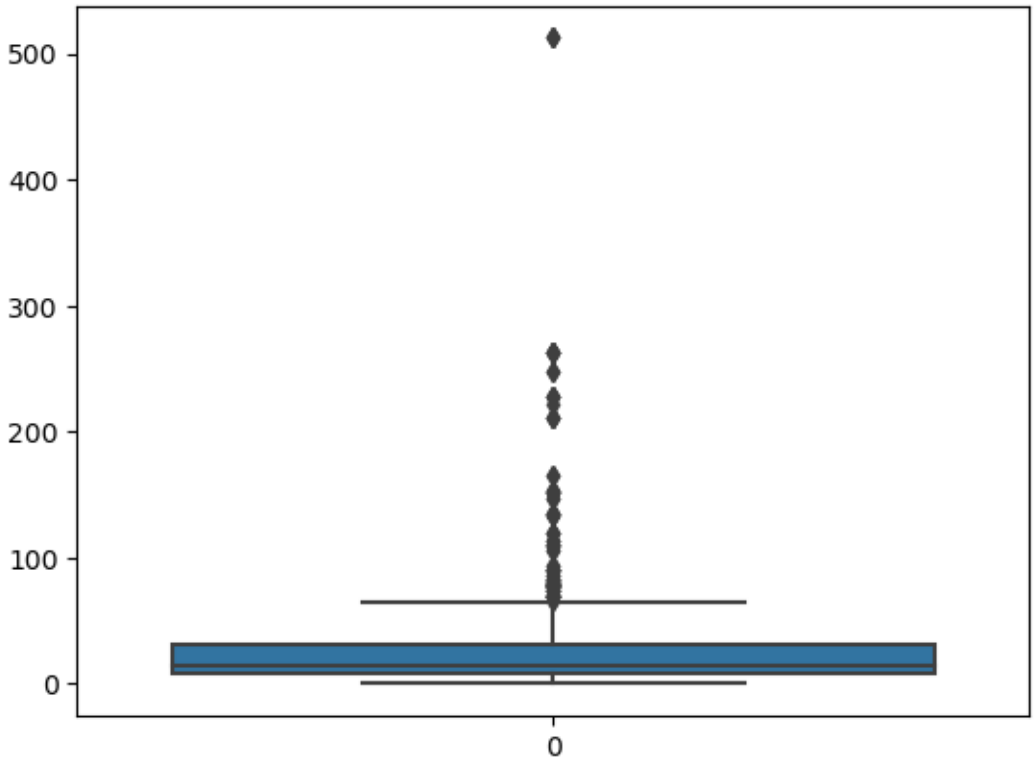
```
adult_male     0
deck           0
embark_town    0
alive          0
alone          0
dtype: int64
```

6. Find the outliers and replace the outliers

1.8 Removing outliers

[38]:
```
sns.boxplot(data.fare)
```

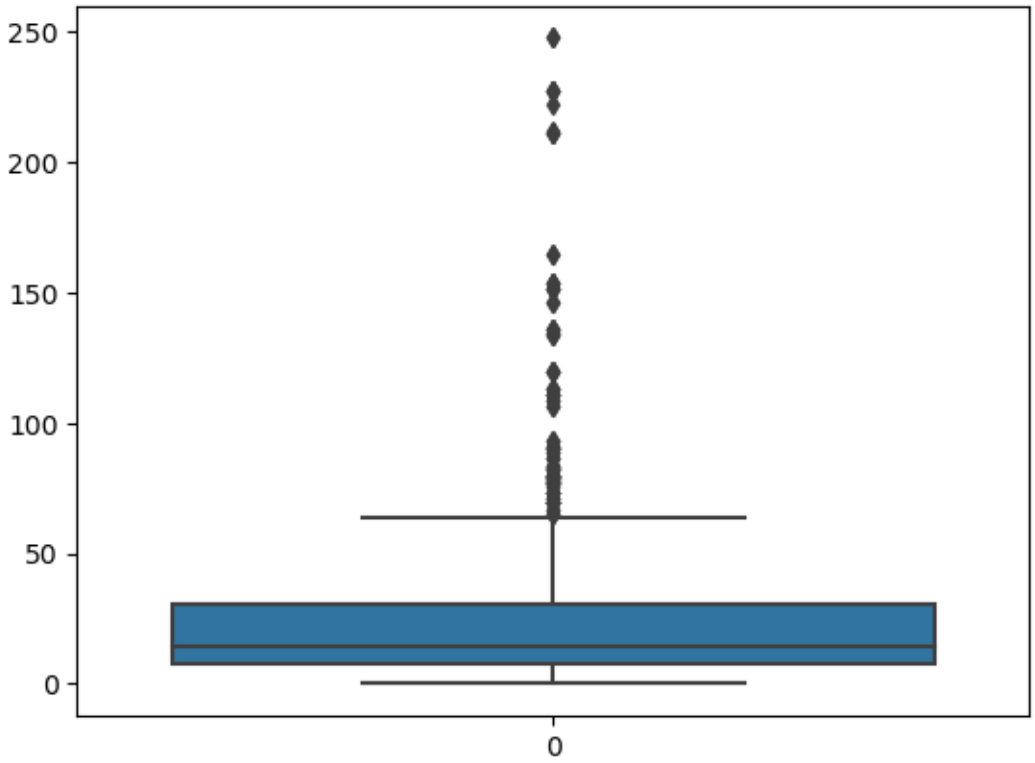
[38]:
```
<Axes: >
```



[39]:
```
quant99 = data.fare.quantile(0.99)
data = data[data.fare<quant99]
```

[40]:
```
sns.boxplot(data.fare)
```

[40]:
```
<Axes: >
```



7. Check for Categorical columns and perform encoding.

1.9 Encoding techniques

1.9.1 Label encoding

[41]:
```
from sklearn.preprocessing import LabelEncoder
```

[42]:
```
le = LabelEncoder()
```

[43]:
```
data.head()
```

[43]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	\
0	0	3	male	22.0	1	0	7.2500	S	Third	
1	1	1	female	38.0	1	0	71.2833	C	First	
2	1	3	female	26.0	0	0	7.9250	S	Third	
3	1	1	female	35.0	1	0	53.1000	S	First	
4	0	3	male	35.0	0	0	8.0500	S	Third	

	who	adult_male	deck	embark_town	alive	alone
0	man	True	C	Southampton	no	False
1	woman	False	C	Cherbourg	yes	False
2	woman	False	C	Southampton	yes	True

```
 3  woman       False    C  Southampton   yes  False
 4    man        True    C  Southampton    no   True
```

[44]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 882 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     882 non-null    int64
 1   pclass       882 non-null    int64
 2   sex          882 non-null    object
 3   age          882 non-null    float64
 4   sibsp        882 non-null    int64
 5   parch        882 non-null    int64
 6   fare         882 non-null    float64
 7   embarked     882 non-null    object
 8   class        882 non-null    object
 9   who          882 non-null    object
 10  adult_male   882 non-null    bool
 11  deck         882 non-null    object
 12  embark_town  882 non-null    object
 13  alive        882 non-null    object
 14  alone        882 non-null    bool
dtypes: bool(2), float64(2), int64(4), object(7)
memory usage: 130.5+ KB
```

[45]:
```
columns = ['sex', 'embarked', 'class', 'who', 'deck', 'alive']
for col in columns:
 data[col] = le.fit_transform(data[col])
```

```
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/462314644.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data[col] = le.fit_transform(data[col])
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/462314644.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data[col] = le.fit_transform(data[col])
```

```
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/462314644.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data[col] = le.fit_transform(data[col])
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/462314644.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data[col] = le.fit_transform(data[col])
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/462314644.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data[col] = le.fit_transform(data[col])
/var/folders/03/k1p5_v6d69bg7b999gdktlgw0000gn/T/ipykernel_3411/462314644.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data[col] = le.fit_transform(data[col])
```

[46]:

```
data.head()
```

[46]:

```
   survived  pclass  sex   age  sibsp  parch     fare  embarked  class  who  \
0         0       3    1  22.0      1      0   7.2500         2      2    1
1         1       1    0  38.0      1      0  71.2833         0      0    2
2         1       3    0  26.0      0      0   7.9250         2      2    2
3         1       1    0  35.0      1      0  53.1000         2      0    2
4         0       3    1  35.0      0      0   8.0500         2      2    1

   adult_male  deck  embark_town  alive  alone
0        True     2  Southampton      0  False
1       False     2    Cherbourg      1  False
2       False     2  Southampton      1   True
3       False     2  Southampton      1  False
```

```
4        True     2  Southampton      0   True
```

1.9.2 One Hot Encoding

[47]:
```
data = pd.get_dummies(data, columns=['embark_town'])
```

[48]:
```
data
```

[48]:
```
     survived  pclass  sex        age  sibsp  parch     fare  embarked  class  \
0           0       3    1  22.000000      1      0   7.2500         2      2
1           1       1    0  38.000000      1      0  71.2833         0      0
2           1       3    0  26.000000      0      0   7.9250         2      2
3           1       1    0  35.000000      1      0  53.1000         2      0
4           0       3    1  35.000000      0      0   8.0500         2      2
..        ...     ...  ...        ...    ...    ...      ...       ...    ...
886         0       2    1  27.000000      0      0  13.0000         2      1
887         1       1    0  19.000000      0      0  30.0000         2      0
888         0       3    0  29.699118      1      2  23.4500         2      2
889         1       1    1  26.000000      0      0  30.0000         0      0
890         0       3    1  32.000000      0      0   7.7500         1      2

     who  adult_male  deck  alive  alone  embark_town_Cherbourg  \
0      1        True     2      0  False                      0
1      2       False     2      1  False                      1
2      2       False     2      1   True                      0
3      2       False     2      1  False                      0
4      1        True     2      0   True                      0
..   ...         ...   ...    ...    ...                    ...
886    1        True     2      0   True                      0
887    2       False     1      1   True                      0
888    2       False     2      0  False                      0
889    1        True     2      1   True                      1
890    1        True     2      0   True                      0

     embark_town_Queenstown  embark_town_Southampton
0                         0                       1
1                         0                       0
2                         0                       1
3                         0                       1
4                         0                       1
..                      ...                     ...
886                       0                       1
887                       0                       1
888                       0                       1
889                       0                       0
890                       1                       0
```

```
[882 rows x 17 columns]
```

8. Split the data into dependent and independent variables.

1.10 Dependent variable

[49]:
```
y = data.loc[:, 'alive':'alive']
y
```

[49]:
```
     alive
0        0
1        1
2        1
3        1
4        0
..     ...
886      0
887      1
888      0
889      1
890      0

[882 rows x 1 columns]
```

1.11 Independent variablea

[50]:
```
X = data.drop(columns=['alive'], axis=1)
X
```

[50]:
```
     survived  pclass  sex        age  sibsp  parch     fare  embarked  class  \
0           0       3    1  22.000000      1      0   7.2500         2      2
1           1       1    0  38.000000      1      0  71.2833         0      0
2           1       3    0  26.000000      0      0   7.9250         2      2
3           1       1    0  35.000000      1      0  53.1000         2      0
4           0       3    1  35.000000      0      0   8.0500         2      2
..        ...     ...  ...        ...    ...    ...      ...       ...    ...
886         0       2    1  27.000000      0      0  13.0000         2      1
887         1       1    0  19.000000      0      0  30.0000         2      0
888         0       3    0  29.699118      1      2  23.4500         2      2
889         1       1    1  26.000000      0      0  30.0000         0      0
890         0       3    1  32.000000      0      0   7.7500         1      2

     who  adult_male  deck  alone  embark_town_Cherbourg  \
0      1        True     2  False                      0
1      2       False     2  False                      1
2      2       False     2   True                      0
3      2       False     2  False                      0
4      1        True     2   True                      0
```

```
..   ...         ...   ...    ...                      ...
886    1        True     2   True                       0
887    2       False     1   True                       0
888    2       False     2  False                       0
889    1        True     2   True                       1
890    1        True     2   True                       0

     embark_town_Queenstown  embark_town_Southampton
0                         0                        1
1                         0                        0
2                         0                        1
3                         0                        1
4                         0                        1
..                      ...                      ...
886                       0                        1
887                       0                        1
888                       0                        1
889                       0                        0
890                       1                        0

[882 rows x 16 columns]
```

9. Scale the independent variables

1.12 Scaling

StandardScaler –> mean=0 std=1 MinMaxScaler –> scale between 0 to 1

[51]:
```
from sklearn.preprocessing import MinMaxScaler
scale = MinMaxScaler()
```

[52]:
```
name = X.columns
X_scaled = scale.fit_transform(X)
```

[53]:
```
X_scaled
```

[53]:
```
array([[0., 1., 1., ..., 0., 0., 1.],
       [1., 0., 0., ..., 1., 0., 0.],
       [1., 1., 0., ..., 0., 0., 1.],
       ...,
       [0., 1., 0., ..., 0., 0., 1.],
       [1., 0., 1., ..., 1., 0., 0.],
       [0., 1., 1., ..., 0., 1., 0.]])
```

[54]:
```
X = pd.DataFrame(X_scaled, columns=name)
X
```

[54]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked
0	0.0	1.0	1.0	0.271174	0.125	0.000000	0.029290	1.0
1	1.0	0.0	0.0	0.472229	0.125	0.000000	0.287989	0.0
2	1.0	1.0	0.0	0.321438	0.000	0.000000	0.032018	1.0
3	1.0	0.0	0.0	0.434531	0.125	0.000000	0.214527	1.0
4	0.0	1.0	1.0	0.434531	0.000	0.000000	0.032523	1.0
..	...	...	...	...	...	...	...	...
877	0.0	0.5	1.0	0.334004	0.000	0.000000	0.052521	1.0
878	1.0	0.0	0.0	0.233476	0.000	0.000000	0.121202	1.0
879	0.0	1.0	0.0	0.367921	0.125	0.333333	0.094740	1.0
880	1.0	0.0	1.0	0.321438	0.000	0.000000	0.121202	0.0
881	0.0	1.0	1.0	0.396833	0.000	0.000000	0.031310	0.5

	class	who	adult_male	deck	alone	embark_town_Cherbourg
0	1.0	0.5	1.0	0.333333	0.0	0.0
1	0.0	1.0	0.0	0.333333	0.0	1.0
2	1.0	1.0	0.0	0.333333	1.0	0.0
3	0.0	1.0	0.0	0.333333	0.0	0.0
4	1.0	0.5	1.0	0.333333	1.0	0.0
..	...	...	...	...	...	...
877	0.5	0.5	1.0	0.333333	1.0	0.0
878	0.0	1.0	0.0	0.166667	1.0	0.0
879	1.0	1.0	0.0	0.333333	0.0	0.0
880	0.0	0.5	1.0	0.333333	1.0	1.0
881	1.0	0.5	1.0	0.333333	1.0	0.0

	embark_town_Queenstown	embark_town_Southampton
0	0.0	1.0
1	0.0	0.0
2	0.0	1.0
3	0.0	1.0
4	0.0	1.0
..	...	...
877	0.0	1.0
878	0.0	1.0
879	0.0	1.0
880	0.0	0.0
881	1.0	0.0

[882 rows x 16 columns]

10. Split the data into training and testing

1.13 Train-Test Split

[55]:
```
from sklearn.model_selection import train_test_split
```

[56]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=0)
```

[57]:
```
X_train
```

[57]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	\
222	1.0	0.0	1.0	0.472229	0.125	0.000000	0.363606	1.0	
200	0.0	1.0	1.0	0.421965	0.000	0.000000	0.026243	1.0	
162	0.0	1.0	1.0	0.007288	0.500	0.166667	0.160340	1.0	
576	0.0	0.5	1.0	0.673285	0.000	0.000000	0.105042	1.0	
841	0.0	1.0	1.0	0.044986	0.500	0.333333	0.126353	1.0	
..	...	...	...	...	...	...	...	...	
835	0.0	1.0	1.0	0.208344	0.000	0.000000	0.034997	1.0	
192	1.0	0.0	0.0	0.547625	0.000	0.000000	0.111994	0.0	
629	1.0	0.5	0.0	0.346569	0.000	0.000000	0.052521	1.0	
559	0.0	1.0	1.0	0.296306	0.250	0.000000	0.097568	1.0	
684	1.0	1.0	0.0	0.044986	0.000	0.166667	0.054204	0.0	

	class	who	adult_male	deck	alone	embark_town_Cherbourg	\
222	0.0	0.5	1.0	0.333333	0.0	0.0	
200	1.0	0.5	1.0	0.333333	1.0	0.0	
162	1.0	0.0	0.0	0.333333	0.0	0.0	
576	0.5	0.5	1.0	0.333333	1.0	0.0	
841	1.0	0.0	0.0	0.333333	0.0	0.0	
..	...	...	...	...	...	...	
835	1.0	0.5	1.0	0.333333	1.0	0.0	
192	0.0	1.0	0.0	0.166667	1.0	1.0	
629	0.5	1.0	0.0	0.333333	1.0	0.0	
559	1.0	0.5	1.0	0.333333	0.0	0.0	
684	1.0	0.0	0.0	0.333333	0.0	1.0	

	embark_town_Queenstown	embark_town_Southampton
222	0.0	1.0
200	0.0	1.0
162	0.0	1.0
576	0.0	1.0
841	0.0	1.0
..	...	...
835	0.0	1.0
192	0.0	0.0
629	0.0	1.0
559	0.0	1.0
684	0.0	0.0

[705 rows x 16 columns]

[58]:
```
y_train
```

[58]:
```
     alive
224      1
202      0
164      0
582      0
850      0
..     ...
844      0
194      1
635      1
565      0
691      1

[705 rows x 1 columns]
```

[59]:
```
X_test
```

[59]:
```
     survived  pclass  sex       age  sibsp     parch      fare  embarked  \
150       0.0     1.0  1.0  0.692134  0.000  0.000000  0.032523       1.0
406       0.0     1.0  1.0  0.367921  0.000  0.000000  0.027708       0.5
513       0.0     1.0  1.0  0.396833  0.000  0.000000  0.031900       1.0
101       0.0     1.0  1.0  0.409399  0.000  0.000000  0.034964       1.0
584       0.0     1.0  1.0  0.434531  0.000  0.000000  0.028785       1.0
..        ...     ...  ...       ...    ...       ...       ...       ...
362       1.0     1.0  0.0  0.367921  0.000  0.000000  0.029206       0.0
367       0.0     1.0  1.0  0.233476  0.000  0.000000  0.032523       1.0
264       1.0     1.0  1.0  0.308872  0.125  0.000000  0.031412       1.0
320       0.0     1.0  1.0  0.367921  1.000  0.333333  0.280986       1.0
466       1.0     0.5  0.0  0.409399  0.125  0.333333  0.112112       1.0

     class  who  adult_male      deck  alone  embark_town_Cherbourg  \
150    1.0  0.5         1.0  0.333333    1.0                    0.0
406    1.0  0.5         1.0  0.333333    1.0                    0.0
513    1.0  0.5         1.0  0.333333    1.0                    0.0
101    1.0  0.5         1.0  0.333333    1.0                    0.0
584    1.0  0.5         1.0  0.333333    1.0                    0.0
..     ...  ...         ...       ...    ...                    ...
362    1.0  1.0         0.0  0.333333    1.0                    1.0
367    1.0  0.5         1.0  0.333333    1.0                    0.0
264    1.0  0.5         1.0  0.333333    0.0                    0.0
320    1.0  0.5         1.0  0.333333    0.0                    0.0
466    0.5  1.0         0.0  0.333333    0.0                    0.0

     embark_town_Queenstown  embark_town_Southampton
150                     0.0                      1.0
406                     1.0                      0.0
513                     0.0                      1.0
```

```
101                   0.0                   1.0
584                   0.0                   1.0
..                    ...                   ...
362                   0.0                   0.0
367                   0.0                   1.0
264                   0.0                   1.0
320                   0.0                   1.0
466                   0.0                   1.0

[177 rows x 16 columns]
```

[60]:
```
y_test
```

[60]:
```
     alive
152      0
411      0
519      0
103      0
590      0
..     ...
367      1
372      0
267      1
324      0
472      1

[177 rows x 1 columns]
```