# **SmartInternz Externships**

## **Applied Data Science**

**Assignment : 2**

**NAME : SAKTHIVEL V**

**REG NO : 20BCE7137**

**Campus: VIT-AP**

In [1]:

**import** numpy **as** np **import** pandas **as** pd **import** seaborn **as** sns

# 2) Load the dataset

In [2]:

df **=** pd**.**read\_csv("titanic.csv")

In [3]:

df**.**head()

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Out[3]: | **survived** | **pclass** | **sex** | **age** | **sibsp** | **parch** | **fare** | **embarked** | **class** | **who** | **adult\_male** | **deck** | **embark\_tow** |
|  | **0** 0 | 3 | male | 22.0 | 1 | 0 | 7.2500 | S | Third | man | True | NaN | Southampto |
|  | **1** 1 | 1 | female | 38.0 | 1 | 0 | 71.2833 | C | First | woman | False | C | Cherbou |
|  | **2** 1 | 3 | female | 26.0 | 0 | 0 | 7.9250 | S | Third | woman | False | NaN | Southampto |
|  | **3** 1 | 1 | female | 35.0 | 1 | 0 | 53.1000 | S | First | woman | False | C | Southampto |
|  | **4** 0 | 3 | male | 35.0 | 0 | 0 | 8.0500 | S | Third | man | True | NaN | Southampto |

In [4]:

<class 'pandas.core.frame.DataFrame'> RangeIndex: 891 entries, 0 to 890 Data columns (total 15 columns):

df**.**info()

# 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

# Perform Below Visualizations.

## Univariate Analysis

* Bi - Variate Analysis

## Multi - Variate Analysis

### univariate analysis

In [5]:

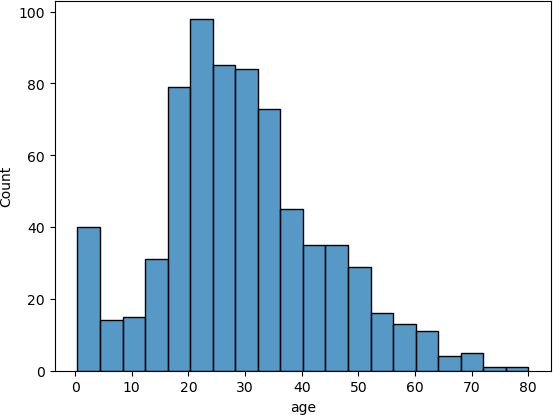
Out[5]:

In [6]:

Out[6]:

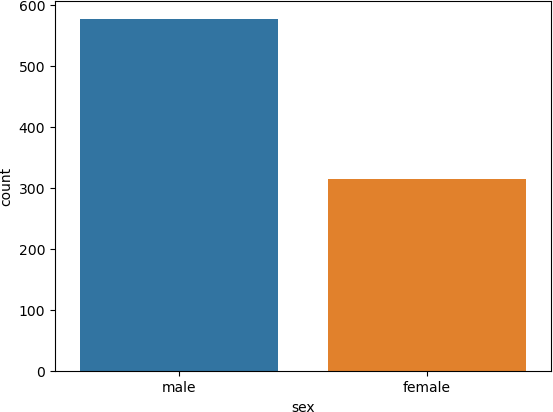
<Axes: xlabel='age', ylabel='Count'>

sns**.**histplot(df['age'])



sns**.**countplot(x **=** df['sex'])

<Axes: xlabel='sex', ylabel='count'>



In [7]:

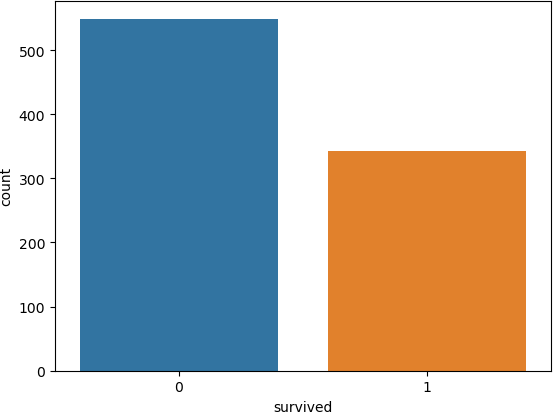
Out[7]:

In [8]:

Out[8]:

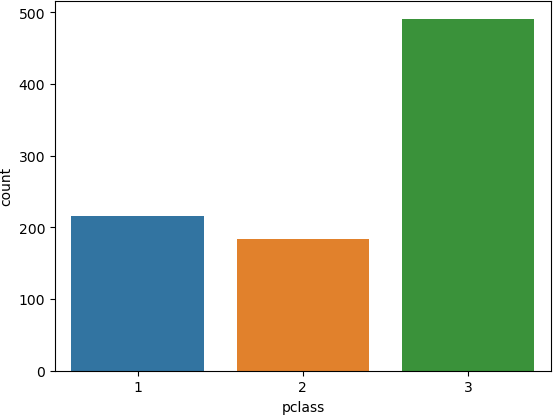
<Axes: xlabel='survived', ylabel='count'>

sns**.**countplot(x **=** df['survived'])



sns**.**countplot(x **=** df['pclass'])

<Axes: xlabel='pclass', ylabel='count'>



In [9]:

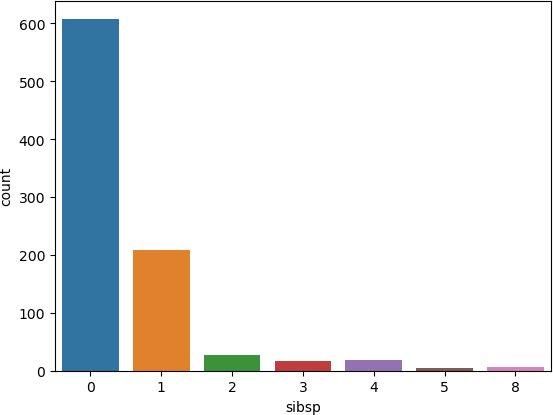
Out[9]:

In [10]:

Out[10]:

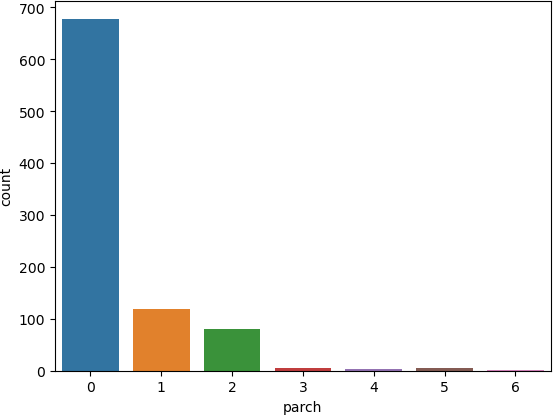
sns**.**countplot(x **=** df['sibsp'])

<Axes: xlabel='sibsp', ylabel='count'>



sns**.**countplot(x **=** df['parch'])

<Axes: xlabel='parch', ylabel='count'>



In [11]:

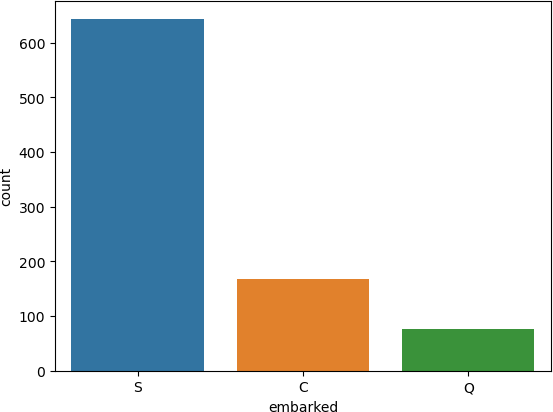
Out[11]:

In [12]:

Out[12]:

sns**.**countplot(x **=** df['embarked'])

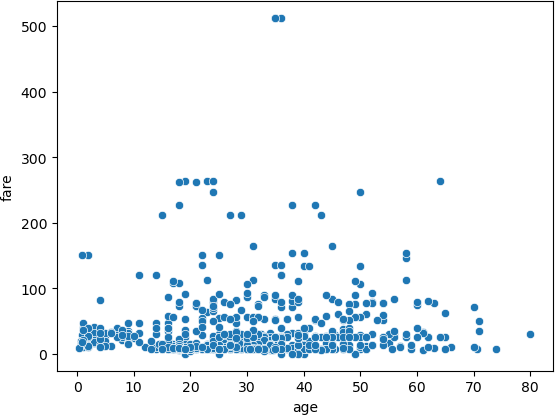
<Axes: xlabel='embarked', ylabel='count'>



### bivariate analysis

sns**.**scatterplot(data **=** df, x **=** 'age', y **=** 'fare')

<Axes: xlabel='age', ylabel='fare'>



In [13]:

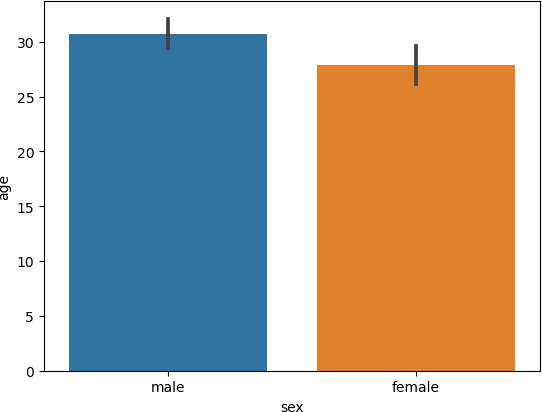
Out[13]:

In [14]:

Out[14]:

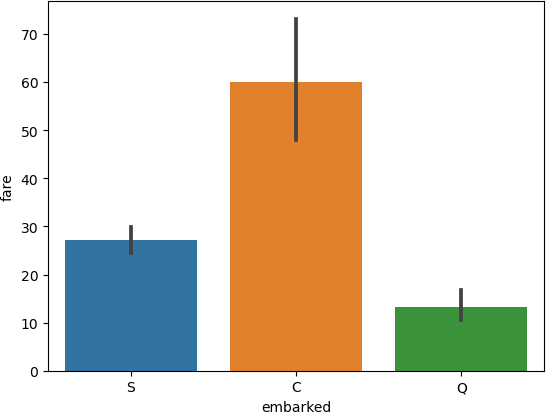
sns**.**barplot(data **=** df, x **=** 'sex', y **=** 'age')

<Axes: xlabel='sex', ylabel='age'>



sns**.**barplot(data **=** df, x **=** 'embarked', y **=** 'fare')

<Axes: xlabel='embarked', ylabel='fare'>



In [15]:

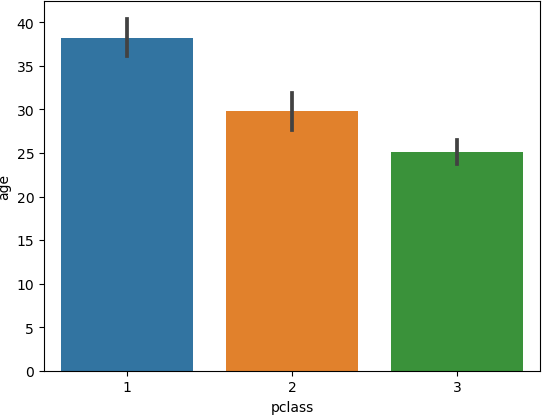
Out[15]:

In [16]:

Out[16]:

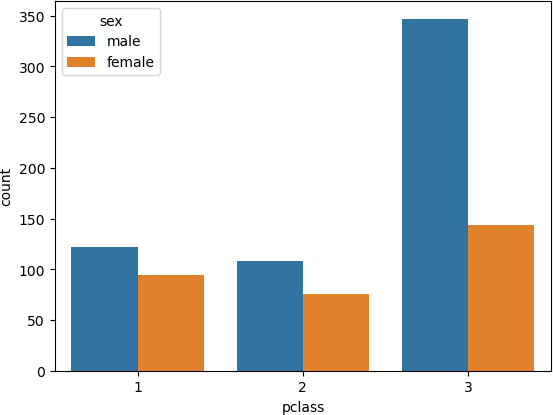
<Axes: xlabel='pclass', ylabel='age'>

sns**.**barplot(data **=** df, x **=** 'pclass', y **=** 'age')



sns**.**countplot(x **=** df['pclass'], hue **=** df['sex'])

<Axes: xlabel='pclass', ylabel='count'>



In [17]:

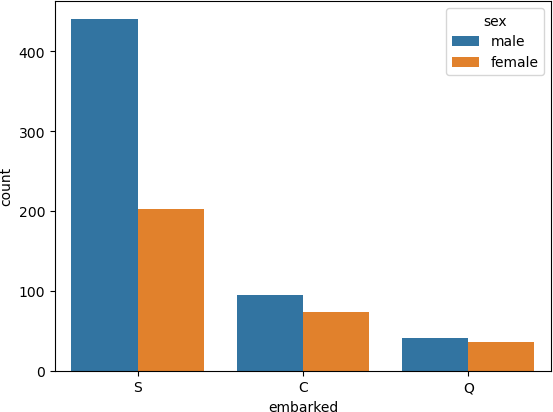
Out[17]:

In [18]:

Out[18]:

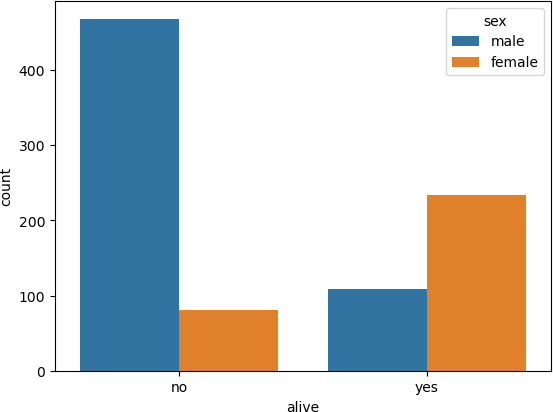
sns**.**countplot(x **=** df['embarked'], hue **=** df['sex'])

<Axes: xlabel='embarked', ylabel='count'>



sns**.**countplot(x **=** df['alive'], hue **=** df['sex'])

<Axes: xlabel='alive', ylabel='count'>



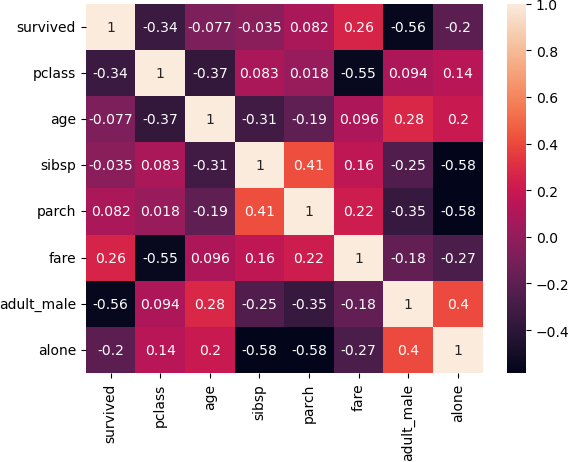
### multivariate analysis

In [19]:

Out[19]:

<Axes: >

sns**.**heatmap(df**.**corr(numeric\_only**=True**), annot **= True**)



# Perform descriptive statistics on the dataset.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| In [20]: | df**.**describe() |  | | | | |
| Out[20]: | **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

# Handle the Missing values.

In [21]:

Out[21]:

In [22]:

df**.**dropna(subset**=**['embark\_town'], how**=**'all', inplace **= True**)

survived 0

df**.**isnull()**.**sum()

|  |  |
| --- | --- |
| 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 |  |

In [23]:

*#for age column we will fill with the average*

df['age'] **=** df['age']**.**fillna(df['age']**.**mean())

In [24]:

*#only 203 records have valid values for deck column so we will drop that*

df**.**drop(['deck'], axis **=** 1,inplace **= True**)

In [25]:

Out[25]:

survived 0

df**.**isnull()**.**sum()

pclass 0

sex 0

age 0

sibsp 0

parch 0

fare 0

embarked 0

class 0

who 0

adult\_male 0

embark\_town 0

alive 0

alone 0

dtype: int64

# Find the outliers and replace the outliers

In [26]:

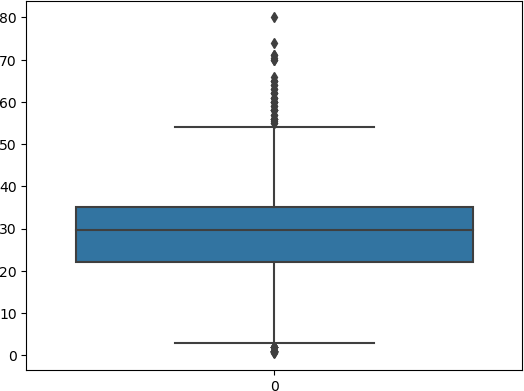
Out[26]:

In [27]:

Out[27]:

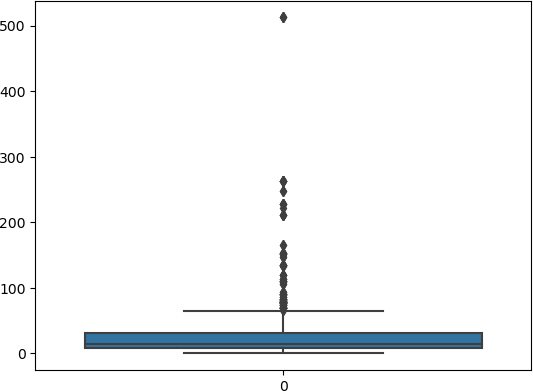
<Axes: >

sns**.**boxplot(df['age'])



sns**.**boxplot(df['fare'])

<Axes: >



In [28]:

median\_age **=** df['age']**.**median()

df["age"] **=** np**.**where(df["age"] **>** 58, median\_age, df['age']) sns**.**boxplot(df['age'])

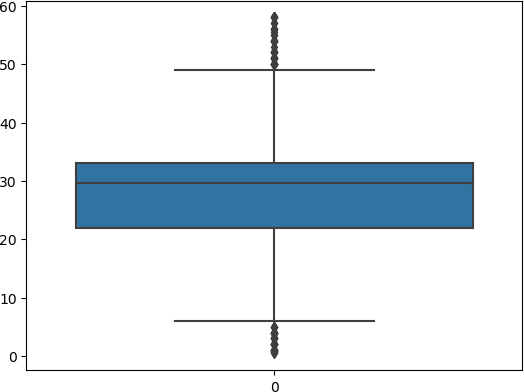
Out[28]:

In [29]:

median\_fare **=** df['fare']**.**median()

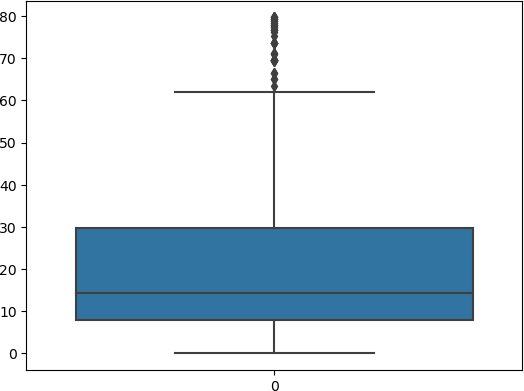
df["fare"] **=** np**.**where(df["fare"] **>** 80, median\_age, df['fare']) sns**.**boxplot(df['fare'])

<Axes: >



Out[29]:

<Axes: >



# Check for Categorical columns and perform encoding.

In [30]:

**from** sklearn.preprocessing **import** OneHotEncoder

In [31]:

encoding **=** pd**.**get\_dummies(df, columns **=** ['sex','embarked','class','who','adult\_male', 'a

In [32]:

encoding**.**head()

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Out[32]: | **survived** | **pclass** | **age** | **sibsp** | **parch** | **fare** | **alive** | **sex\_female** | **sex\_male** | **embarked\_C ... who\_child who\_m** |
| **0** | 0 | 3 | 22.0 | 1 | 0 | 7.2500 | no | 0 | 1 | 0 ... 0 |
| **1** | 1 | 1 | 38.0 | 1 | 0 | 71.2833 | yes | 1 | 0 | 1 ... 0 |
| **2** | 1 | 3 | 26.0 | 0 | 0 | 7.9250 | yes | 1 | 0 | 0 ... 0 |
| **3** | 1 | 1 | 35.0 | 1 | 0 | 53.1000 | yes | 1 | 0 | 0 ... 0 |
| **4** | 0 | 3 | 35.0 | 0 | 0 | 8.0500 | no | 0 | 1 | 0 ... 0 |

5 rows × 25 columns

# Split the data into dependent and independent variables

In [33]:

Out[33]:

Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare', 'embarked', 'class', 'who', 'adult\_male', 'embark\_town', 'alive',

df**.**columns

'alone'], dtype='object')

In [34]:

*# independent variables*

X **=** encoding**.**drop(['survived', 'alive'], axis **=** 1) X**.**head()

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Out[34]: | **pclass** | **age** | **sibsp** | **parch** | **fare** | **sex\_female** | **sex\_male** | **embarked\_C** | **embarked\_Q** | **embarked\_S** | **... who\_chi** |
| **0** | 3 | 22.0 | 1 | 0 | 7.2500 | 0 | 1 | 0 | 0 | 1 | ... |
| **1** | 1 | 38.0 | 1 | 0 | 71.2833 | 1 | 0 | 1 | 0 | 0 | ... |
| **2** | 3 | 26.0 | 0 | 0 | 7.9250 | 1 | 0 | 0 | 0 | 1 | ... |
| **3** | 1 | 35.0 | 1 | 0 | 53.1000 | 1 | 0 | 0 | 0 | 1 | ... |
| **4** | 3 | 35.0 | 0 | 0 | 8.0500 | 0 | 1 | 0 | 0 | 1 | ... |

### 5 rows × 23 columns

In [35]:

*# dependent variables*

y **=** df[['survived', 'alive']] y**.**head()

|  |  |  |  |
| --- | --- | --- | --- |
| Out[35]: |  | **survived** | **alive** |
|  | **0** | 0 | no |
|  | **1** | 1 | yes |
|  | **2** | 1 | yes |
|  | **3** | 1 | yes |
|  | **4** | 0 | no |

# Scaling the independent variables

In [36]:

**from** sklearn.preprocessing **import** StandardScaler scaler **=** StandardScaler()

x\_std **=** scaler**.**fit\_transform(X)

In [37]:

Out[37]:

array([[ 0.82520863, -0.57985934, 0.43135024, ..., -0.48271079,

x\_std

|  |  |  |  |
| --- | --- | --- | --- |
| -0.30794088, | 0.61679395], |  | |
| [-1.57221121, | 0.83108889, 0.43135024, | ..., | 2.07163382, |
| -0.30794088, | -1.62128697], |  |  |
| [ 0.82520863, | -0.22712228, -0.47519908, | ..., | -0.48271079, |
| -0.30794088, | 0.61679395], |  |  |
| ..., |  |  |  |
| [ 0.82520863, | 0.09405298, 0.43135024, | ..., | -0.48271079, |
| -0.30794088, | 0.61679395], |  |  |
| [-1.57221121, | -0.22712228, -0.47519908, | ..., | 2.07163382, |
| -0.30794088, | -1.62128697], |  |  |
| [ 0.82520863, | 0.3019833 , -0.47519908, | ..., | -0.48271079, |
| 3.24737656, | -1.62128697]]) |  |  |

# 10.Split the data into training and testing

**from** sklearn.model\_selection **import** train\_test\_split

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y['survived'], test\_size**=**0.33, ra

In [38]:

In [ ]: