In [1]:
```
#reading the data
import pandas as  pd
titanic_data= pd.read_csv("titanic_dataset.csv")
```

In [2]:
```
titanic_data.head()
```

Out[2]:

| | PassengerId | Survived | Pclass | Name | Gender | Age | SibSp | Parch | Ticket | Fa |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22.0 | 1 | 0 | A/5 21171 | 7.25 |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 | 71.28 |
| **2** | 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.92 |
| **3** | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35.0 | 1 | 0 | 113803 | 53.10 |
| **4** | 5 | 0 | 3 | Allen, Mr. William Henry | male | 35.0 | 0 | 0 | 373450 | 8.05 |

In [3]:
```
#remove case sensitive (capital/small)
titanic_data.columns = [cols.lower() for cols in titanic_data.colum
```

In [4]:
```
titanic_data.head()
```

Out[4]:

| | passengerid | survived | pclass | name | gender | age | sibsp | parch | ticket | far |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22.0 | 1 | 0 | A/5 21171 | 7.250 |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 | 71.283 |
| **2** | 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.925 |
| **3** | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35.0 | 1 | 0 | 113803 | 53.100 |
| **4** | 5 | 0 | 3 | Allen, Mr. William Henry | male | 35.0 | 0 | 0 | 373450 | 8.050 |

In [5]:
```
#check wide information
titanic_data.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   gender       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
```

In [7]:

```
#check null value
titanic_data.isna().sum()
```

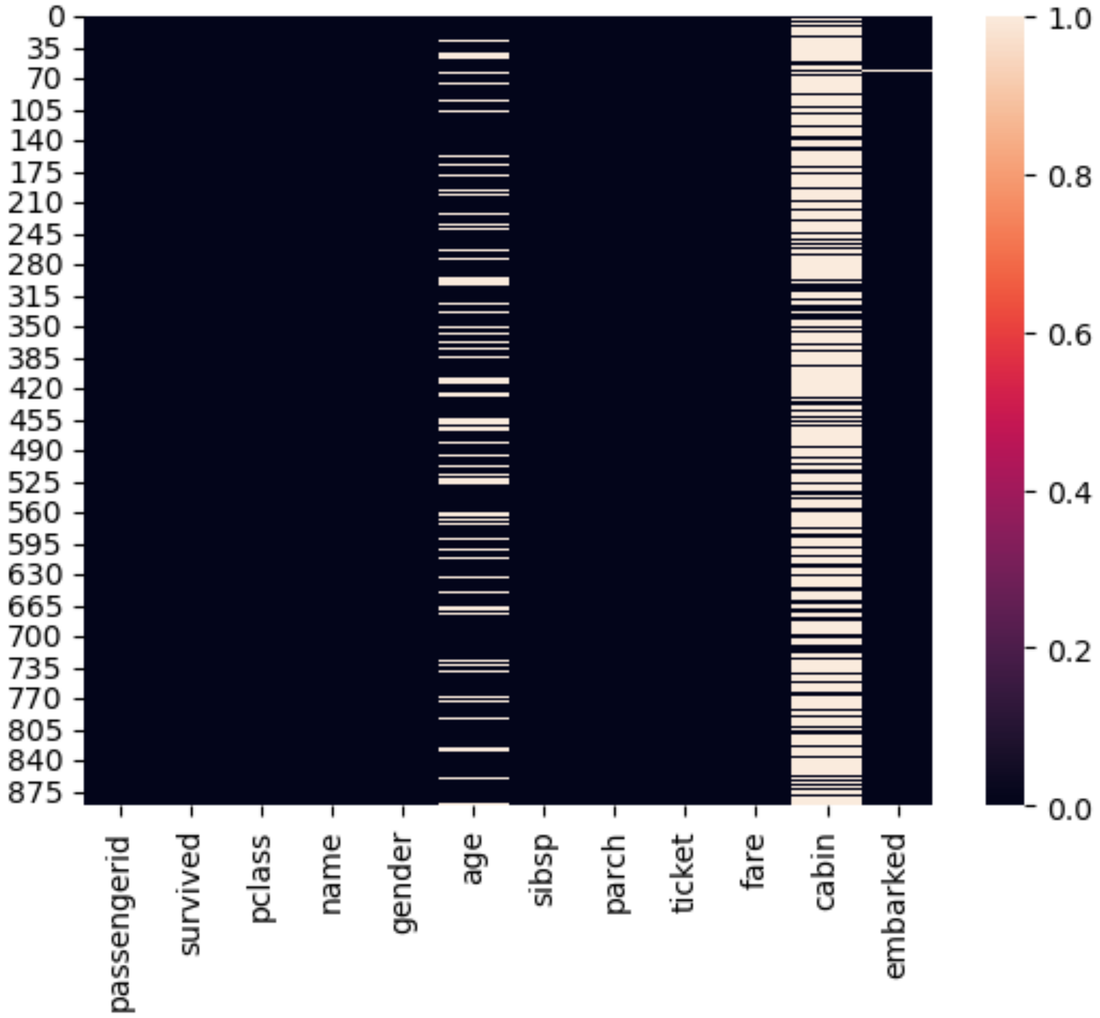
Out[7]:

```
passengerid      0
survived         0
pclass           0
name             0
gender           0
age            177
sibsp            0
parch            0
ticket           0
fare             0
cabin          687
embarked         2
dtype: int64
```

# heatmap

In [8]:
```
import seaborn as sea
sea.heatmap(titanic_data.isna())
```

Out[8]:
```
<Axes: >
```



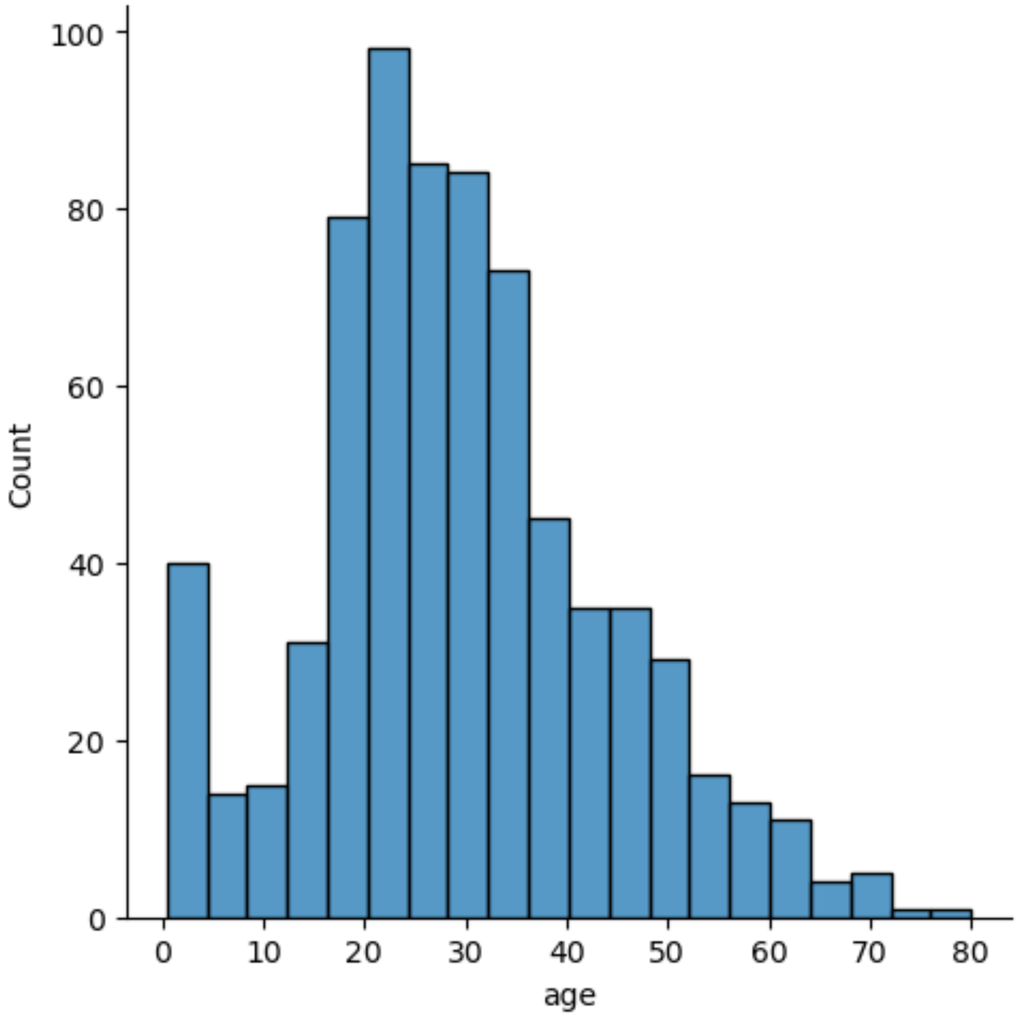
In [9]:

```
sea.displot(x="age" , data = titanic_data)
```

```
/opt/homebrew/anaconda3/lib/python3.11/site-packages/seaborn/_oldc
ore.py:1498: FutureWarning: is_categorical_dtype is deprecated and
will be removed in a future version. Use isinstance(dtype, Categor
icalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/homebrew/anaconda3/lib/python3.11/site-packages/seaborn/_oldc
ore.py:1119: FutureWarning: use_inf_as_na option is deprecated and
will be removed in a future version. Convert inf values to NaN bef
ore operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/homebrew/anaconda3/lib/python3.11/site-packages/seaborn/axisg
rid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[9]:

```
<seaborn.axisgrid.FacetGrid at 0x117cd4250>
```



# mean age calculate (because age column free space fill)

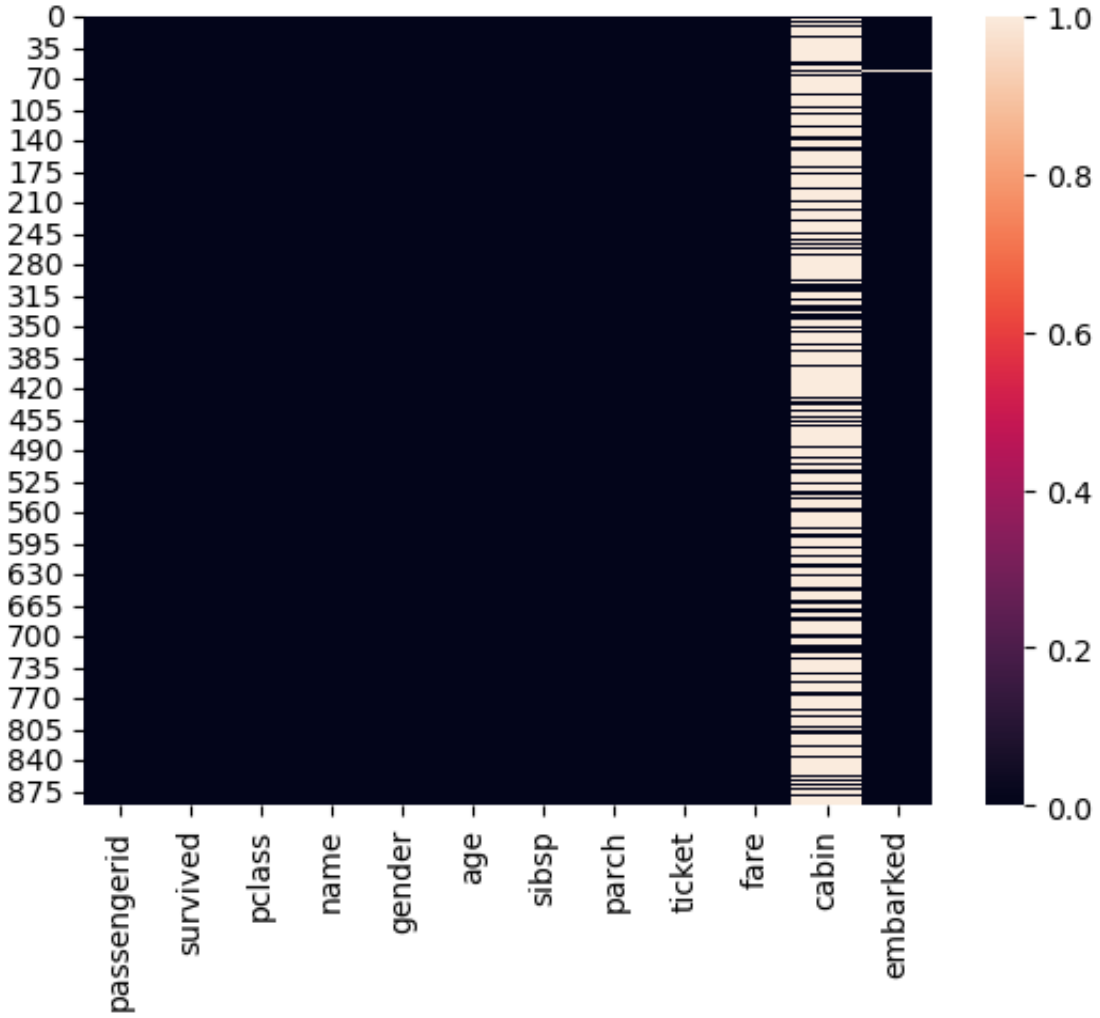
In [11]:
```
mean_age = titanic_data["age"].mean()
print(mean_age)
```

```
29.69911764705882
```

In [12]:
```
titanic_data ["age"].fillna(mean_age,inplace = True)
```

In [13]:
```
sea.heatmap(titanic_data.isna())
```

Out[13]:
```
<Axes: >
```



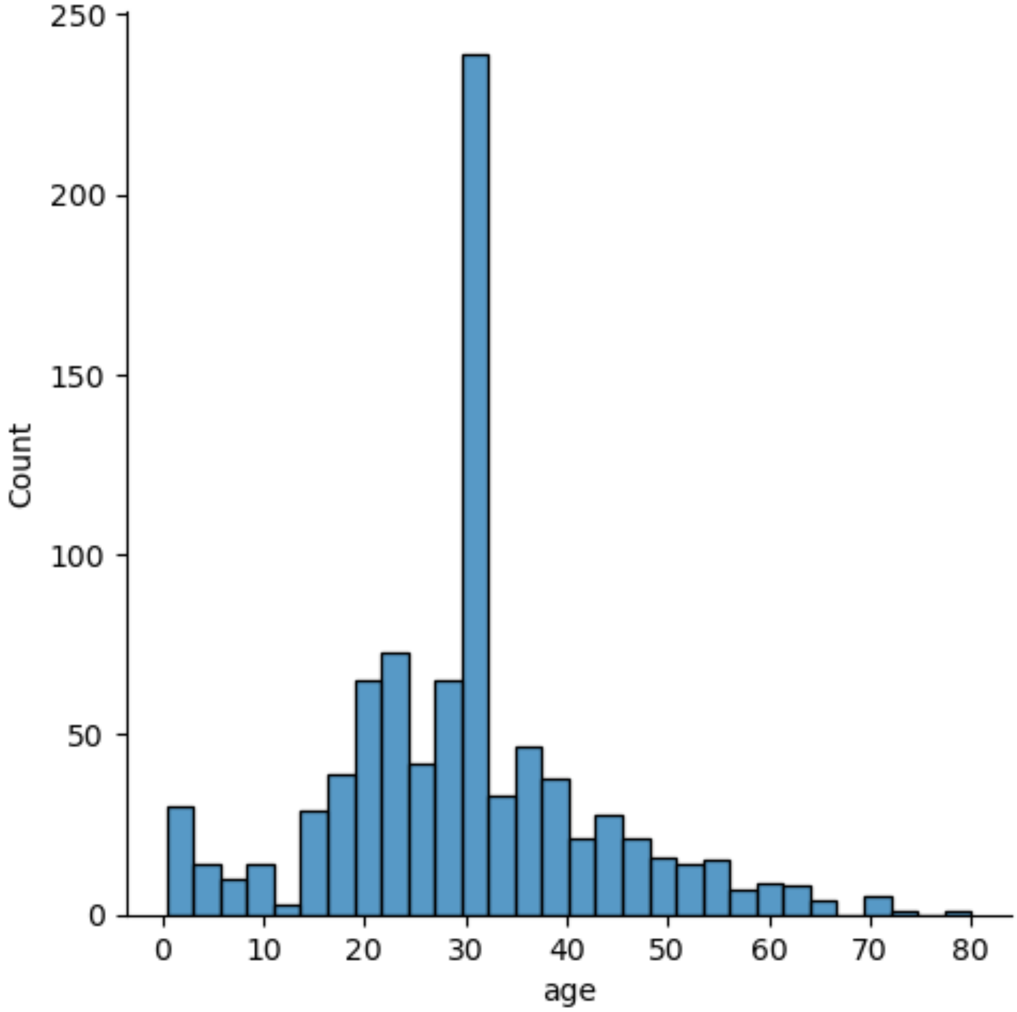
In [14]:

```
sea.displot(x="age" , data = titanic_data)
```

```
/opt/homebrew/anaconda3/lib/python3.11/site-packages/seaborn/_oldc
ore.py:1498: FutureWarning: is_categorical_dtype is deprecated and
will be removed in a future version. Use isinstance(dtype, Categor
icalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/homebrew/anaconda3/lib/python3.11/site-packages/seaborn/_oldc
ore.py:1119: FutureWarning: use_inf_as_na option is deprecated and
will be removed in a future version. Convert inf values to NaN bef
ore operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/homebrew/anaconda3/lib/python3.11/site-packages/seaborn/axisg
rid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[14]:

```
<seaborn.axisgrid.FacetGrid at 0x15623b8d0>
```



```
In [15]: titanic_data.isna().sum()
```

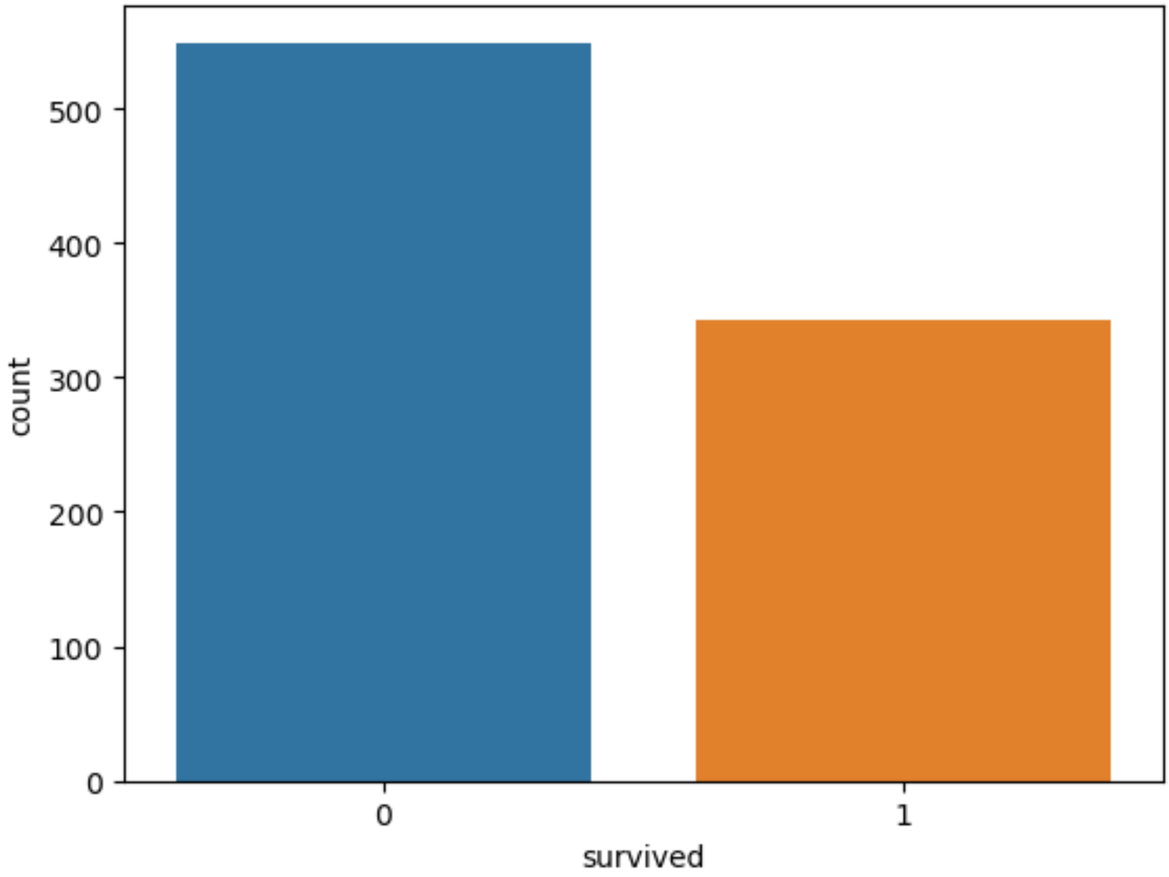
```
Out[15]: passengerid      0
         survived         0
         pclass           0
         name             0
         gender           0
         age              0
         sibsp            0
         parch            0
         ticket           0
         fare             0
         cabin          687
         embarked         2
         dtype: int64
```

```
In [ ]:
```

In [19]:
```
sea.countplot(x = "survived" ,data = titanic_data)
```

```
/opt/homebrew/anaconda3/lib/python3.11/site-packages/seaborn/_oldc
ore.py:1498: FutureWarning: is_categorical_dtype is deprecated and
will be removed in a future version. Use isinstance(dtype, Categor
icalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/homebrew/anaconda3/lib/python3.11/site-packages/seaborn/_oldc
ore.py:1498: FutureWarning: is_categorical_dtype is deprecated and
will be removed in a future version. Use isinstance(dtype, Categor
icalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/homebrew/anaconda3/lib/python3.11/site-packages/seaborn/_oldc
ore.py:1498: FutureWarning: is_categorical_dtype is deprecated and
will be removed in a future version. Use isinstance(dtype, Categor
icalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

Out[19]:
```
<Axes: xlabel='survived', ylabel='count'>
```

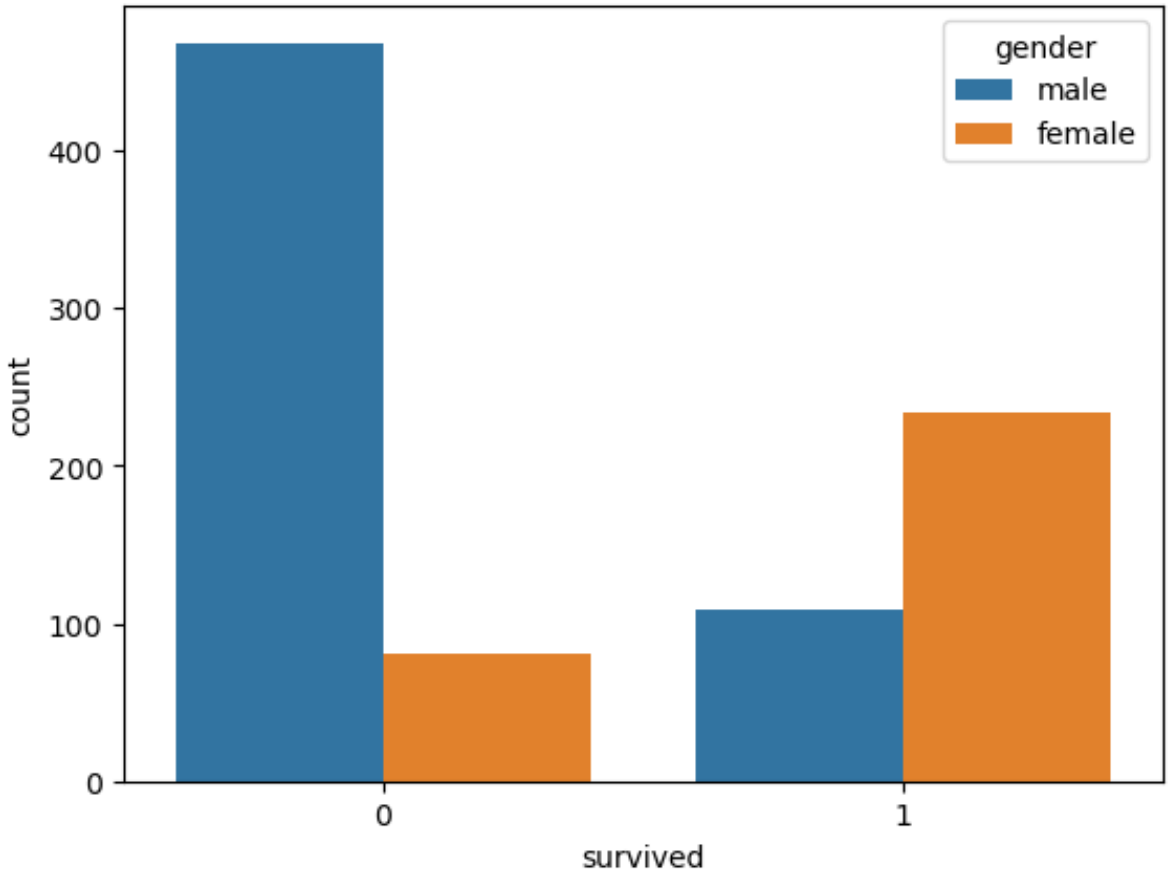


# check male,female survived data

In [20]:
```
sea.countplot(x = "survived" ,data = titanic_data , hue ="gender")
```

```
/opt/homebrew/anaconda3/lib/python3.11/site-packages/seaborn/_oldc
ore.py:1498: FutureWarning: is_categorical_dtype is deprecated and
will be removed in a future version. Use isinstance(dtype, Categor
icalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/homebrew/anaconda3/lib/python3.11/site-packages/seaborn/_oldc
ore.py:1498: FutureWarning: is_categorical_dtype is deprecated and
will be removed in a future version. Use isinstance(dtype, Categor
icalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/homebrew/anaconda3/lib/python3.11/site-packages/seaborn/_oldc
ore.py:1498: FutureWarning: is_categorical_dtype is deprecated and
will be removed in a future version. Use isinstance(dtype, Categor
icalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/homebrew/anaconda3/lib/python3.11/site-packages/seaborn/_oldc
ore.py:1498: FutureWarning: is_categorical_dtype is deprecated and
will be removed in a future version. Use isinstance(dtype, Categor
icalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

Out[20]:
```
<Axes: xlabel='survived', ylabel='count'>
```



# convert letter to numerial value

(assign the numbers eg: male-1,female-2,......)

In [21]:
```
titanic_data.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   gender       891 non-null    object 
 5   age          891 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
```

In [22]:
```
print(titanic_data["gender"])
```

```
0        male
1      female
2      female
3      female
4        male
        ...  
886      male
887    female
888    female
889      male
890      male
Name: gender, Length: 891, dtype: object
```

In [24]:
```
titanic_data.replace({"gender":{"male":1, "female":2 }}, inplace =T
```

In [25]:

```
print(titanic_data["gender"])
```

```
0      1
1      2
2      2
3      2
4      1
      ..
886    1
887    2
888    2
889    1
890    1
Name: gender, Length: 891, dtype: int64
```

In [26]:

```
titanic_data.describe()
```

Out[26]:

| | passengerid | survived | pclass | gender | age | sibsp | parc |
|---|---|---|---|---|---|---|---|
| **count** | 891.000000 | 891.000000 | 891.000000 | 891.000000 | 891.000000 | 891.000000 | 891.00000 |
| **mean** | 446.000000 | 0.383838 | 2.308642 | 1.352413 | 29.699118 | 0.523008 | 0.38159 |
| **std** | 257.353842 | 0.486592 | 0.836071 | 0.477990 | 13.002015 | 1.102743 | 0.80605 |
| **min** | 1.000000 | 0.000000 | 1.000000 | 1.000000 | 0.420000 | 0.000000 | 0.00000 |
| **25%** | 223.500000 | 0.000000 | 2.000000 | 1.000000 | 22.000000 | 0.000000 | 0.00000 |
| **50%** | 446.000000 | 0.000000 | 3.000000 | 1.000000 | 29.699118 | 0.000000 | 0.00000 |
| **75%** | 668.500000 | 1.000000 | 3.000000 | 2.000000 | 35.000000 | 1.000000 | 0.00000 |
| **max** | 891.000000 | 1.000000 | 3.000000 | 2.000000 | 80.000000 | 8.000000 | 6.00000 |

In [27]:

```
titanic_data.columns
```

Out[27]:

```
Index(['passengerid', 'survived', 'pclass', 'name', 'gender', 'ag
e', 'sibsp',
       'parch', 'ticket', 'fare', 'cabin', 'embarked'],
      dtype='object')
```

In [37]:

```
input_data = titanic_data.drop(columns = ["name"  ,"ticket", "embar
output_data = titanic_data ["survived"]
```

In [38]:
```
print (input_data)
```

```
     passengerid  pclass  gender        age  sibsp  parch     fare
0              1       3       1  22.000000      1      0   7.2500
1              2       1       2  38.000000      1      0  71.2833
2              3       3       2  26.000000      0      0   7.9250
3              4       1       2  35.000000      1      0  53.1000
4              5       3       1  35.000000      0      0   8.0500
..           ...     ...     ...        ...    ...    ...      ...
886          887       2       1  27.000000      0      0  13.0000
887          888       1       2  19.000000      0      0  30.0000
888          889       3       2  29.699118      1      2  23.4500
889          890       1       1  26.000000      0      0  30.0000
890          891       3       1  32.000000      0      0   7.7500

[891 rows x 7 columns]
```

In [39]:
```
print (output_data)
```

```
0      0
1      1
2      1
3      1
4      0
      ..
886    0
887    1
888    0
889    1
890    0
Name: survived, Length: 891, dtype: int64
```

In [ ]:

# change number simliar range of number (StandardScaler)

In [40]:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
input_data = scaler.fit_transform(input_data)
print (input_data)
```

```
[[-1.73010796  0.82737724 -0.73769513 ...  0.43279337 -0.47367361
  -0.50244517]
 [-1.72622007 -1.56610693  1.35557354 ...  0.43279337 -0.47367361
   0.78684529]
 [-1.72233219  0.82737724  1.35557354 ... -0.4745452  -0.47367361
  -0.48885426]
 ...
 [ 1.72233219  0.82737724  1.35557354 ...  0.43279337  2.00893337
  -0.17626324]
 [ 1.72622007 -1.56610693 -0.73769513 ... -0.4745452  -0.47367361
  -0.04438104]
 [ 1.73010796  0.82737724 -0.73769513 ... -0.4745452  -0.47367361
  -0.49237783]]
```

In [ ]:

In [41]:

```
print(input_data.shape)
print(output_data.shape)
```

```
(891, 7)
(891,)
```

In [ ]:

# Model test (allocation datas test,train)

In [57]:

```
from sklearn.model_selection import train_test_split
input_data_train , input_data_test , output_data_train , output_dat
```

In [58]:

```
print(input_data_train.shape)
print(output_data_train.shape)
print(input_data_test.shape)
print(output_data_test.shape)
```

```
(712, 7)
(712,)
(179, 7)
(179,)
```

In [ ]:

# Model bulid

In [59]:

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(input_data_train , output_data_train)
```

Out[59]:

```
LogisticRegression()
```

**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.**
**On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

In [ ]:

# predict the data

In [60]:

```
predited_survived = model.predict(input_data_test)
```

In [61]:

```
predited_survived
```

Out[61]:

```
array([1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0,
       0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1,
1, 1,
       0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0, 0,
       0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
0, 1,
       0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1,
0, 1,
       0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0,
       1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0,
1, 0,
       0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,
0, 1,
       0, 0, 0])
```

In [ ]:

# compare predit value and assign value using confusion_matrix

In [62]:

```
from sklearn.metrics import confusion_matrix
```

In [63]:

```
confusion_matrix(output_data_test , predited_survived)
```

Out[63]:

```
array([[92, 14],
       [20, 53]])
```

In [64]:

```
pd.DataFrame(confusion_matrix(output_data_test , predited_survived)
```

Out[64]:

| | predited No | predited Yes |
|---|---|---|
| **Actual No** | 92 | 14 |
| **Actual Yes** | 20 | 53 |

In [ ]:

In [ ]:

# accuracy check

In [65]:

```
from sklearn.metrics import accuracy_score
accuracy_info = accuracy_score(output_data_test , predited_survived
```

In [66]:

```
print(accuracy_info)
```

```
0.8100558659217877
```

In [ ]:

In [68]:

```
from sklearn.linear_model import LogisticRegression
import joblib
titanic_model = LogisticRegression()
titanic_model.fit(input_data,output_data)
joblib.dump(titanic_model , "surviver_predited_Identifier")
```

Out[68]:

```
['surviver_predited_Identifier']
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: