

SAMIR SHAH

Samir Shah

VIT VELLORE

In

[64]:

In

[65]:

**import**

pandas

**as**

pd

**import**

numpy

**as**

np

**import**

matplotlib

.

pyplot

**as**

plt

**import**

seaborn

**as**

sns

**%**

matplotlib

inline

1

2

3

4

5

data

**=**

pd

.

read\_csv

(

'titanic.csv'

)

1

In [66]:

data

.

head

()

1

Out[66]:

**PassengerId Survived Pclass Name Sex Age SibSp Parch Ticket Fare**

Kelly, Mr.

1. 892 0 3 male 34.5 0 0 330911 7.8292

James

Wilkes, Mrs.

1. 893 1 3 James female 47.0 1 0 363272 7.0000

(Ellen Needs)

Myles, Mr.

1. 894 0 2 male 62.0 0 0 240276 9.6875

Thomas

Francis

Wirz, Mr.

1. 895 0 3 male 27.0 0 0 315154 8.6625

Albert

Hirvonen, Mrs.

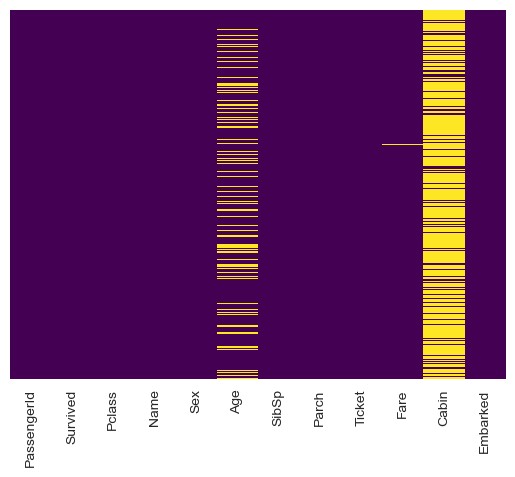
1. 896 1 3 Alexander female 22.0 1 1 3101298 12.2875

(Helga E Lindqvist)

[67]: sns.heatmap(data.isnull(),yticklabels**=False**,cbar**=False**,cmap**=**'viridis')

Out[67]:

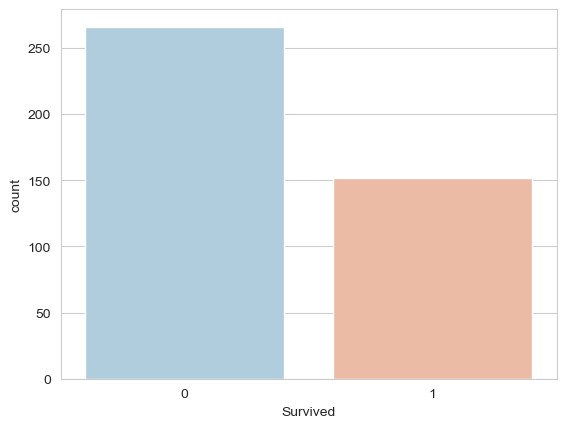
<Axes: >



[68]: sns.countplot(x**=**'Survived',data**=**data,palette**=**'RdBu\_r')

Out[68]:

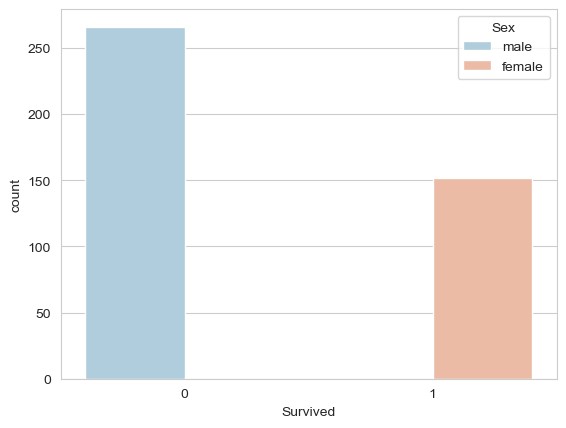
<Axes: xlabel='Survived', ylabel='count'>



[69]: sns.countplot(x**=**'Survived',hue**=**'Sex',data**=**data,palette**=**'RdBu\_r')

Out[69]:

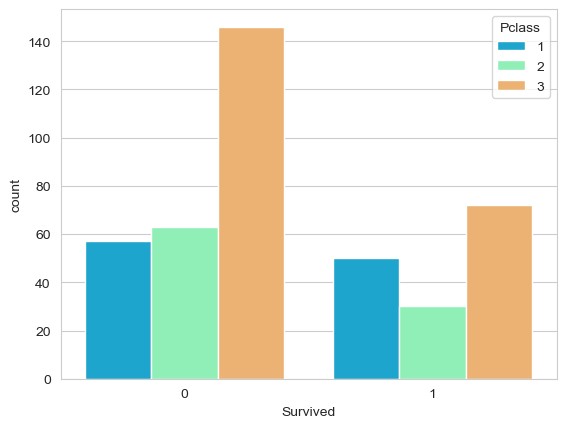
<Axes: xlabel='Survived', ylabel='count'>



[70]: sns.countplot(x**=**'Survived',hue**=**'Pclass',data**=**data,palette**=**'rainbow')

Out[70]:

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



[33]: sns.distplot(data['Age'].dropna(),kde**=False**,color**=**'darkred',bins**=**30)

C:\Users\suraj\AppData\Local\Temp\ipykernel\_10448\4202982456.py:1: UserWa rning:

`distplot` is a deprecated function and will be removed in seaborn v0.14. 0.

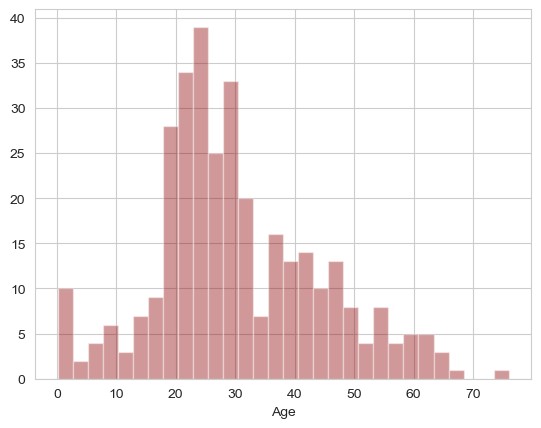
Please adapt your code to use either `displot` (a figure-level function w ith

similar flexibility) or `histplot` (an axes-level function for histogram s).

For a guide to updating your code to use the new functions, please see [https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (http s://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)](https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)  sns.distplot(data['Age'].dropna(),kde=False,color='darkred',bins=30)

Out[33]:

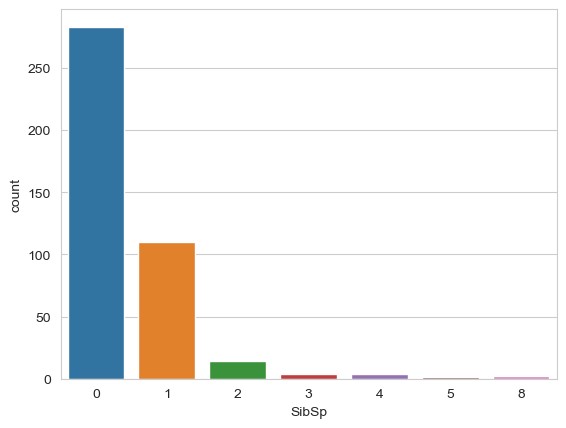
<Axes: xlabel='Age'>



[35]: sns.countplot(x**=**'SibSp',data**=**data)

Out[35]:

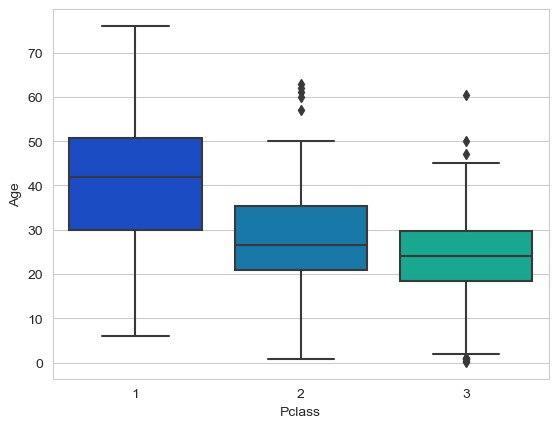
<Axes: xlabel='SibSp', ylabel='count'>



[36]: sns.boxplot(x**=**'Pclass',y**=**'Age',data**=**data,palette**=**'winter')

Out[36]:

<Axes: xlabel='Pclass', ylabel='Age'>



In

[38]:

**def**

impute\_age

(

cols

):

Age

**=**

cols

[

0

]

Pclass

**=**

cols

[

1

]

**if**

pd

.

isnull

(

Age

):

**if**

Pclass

**==**

1

:

**return**

37

**elif**

Pclass

**==**

2

:

**return**

29

**else**

:

**return**

24

**else**

:

**return**

Age

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

[39]: data['Age'] **=** data[['Age','Pclass']].apply(impute\_age,axis**=**1)

In [40]:

sns

.

heatmap

(

data

.

isnull

()

,

yticklabels

**=**

**False**

,

cbar

**=**

**False**

,

cmap

**=**

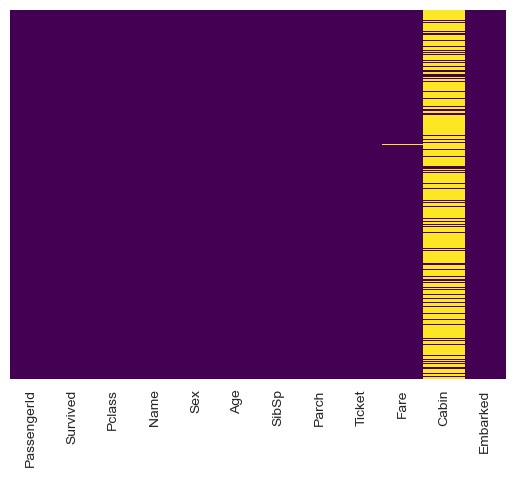
'viridis'

)

1

Out[40]:

<Axes: >



In [41]:

data

.

drop

(

'Cabin'

,

axis

**=**

1

,

inplace

**=**

**True**

)

1

[42]: data.head()

Out[42]:

**PassengerId Survived Pclass Name Sex Age SibSp Parch Ticket Fare**

Kelly, Mr.

1. 892 0 3 male 34.5 0 0 330911 7.8292

James

Wilkes, Mrs.

1. 893 1 3 James female 47.0 1 0 363272 7.0000

(Ellen Needs)

Myles, Mr.

1. 894 0 2 male 62.0 0 0 240276 9.6875

Thomas

Francis

Wirz, Mr.

1. 895 0 3 male 27.0 0 0 315154 8.6625

Albert

Hirvonen, Mrs.

1. 896 1 3 Alexander female 22.0 1 1 3101298 12.2875

(Helga E

Lindqvist)

In [43]:

data

.

dropna

(

inplace

**=**

**True**

)

1

In [44]:

data

.

info

()

1

<class 'pandas.core.frame.DataFrame'>

Int64Index: 417 entries, 0 to 417 Data columns (total 11 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

1. PassengerId 417 non-null int64
2. Survived 417 non-null int64
3. Pclass 417 non-null int64
4. Name 417 non-null object
5. Sex 417 non-null object
6. Age 417 non-null float64
7. SibSp 417 non-null int64
8. Parch 417 non-null int64
9. Ticket 417 non-null object
10. Fare 417 non-null float64
11. Embarked 417 non-null object dtypes: float64(2), int64(5), object(4) memory usage: 39.1+ KB

[45]:

1. sex **=** pd.get\_dummies(data['Sex'],drop\_first**=True**)
2. embark **=** pd.get\_dummies(data['Embarked'],drop\_first**=True**) In [46]:

data

.

drop

([

'Sex'

,

'Embarked'

,

'Name'

,

'Ticket'

]

,

axis

**=**

1

,

inplace

**=**

**True**

)

1

In [47]:

data

**=**

pd

.

concat

([

data

,

sex

,

embark

]

,

axis

**=**

1

)

1

In [48]:

data

.

head

()

1

Out[48]:

**PassengerId Survived Pclass Age SibSp Parch Fare male Q S**

1. 892 0 3 34.5 0 0 7.8292 1 1 0

lm

**=**

LinearRegression

()

1

1. 893 1 3 47.0 1 0 7.0000 0 0 1
2. 894 0 2 62.0 0 0 9.6875 1 1 0
3. 895 0 3 27.0 0 0 8.6625 1 0 1
4. 896 1 3 22.0 1 1 12.2875 0 0 1

In [49]:

**from**

sklearn

.

model\_selection

**import**

train\_test\_split

1

In [51]:

1. X **=** train.drop('Survived',axis**=**1)
2. y **=** train['Survived']
3. X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=**0.3, random\_stat In [52]:

**from**

sklearn

.

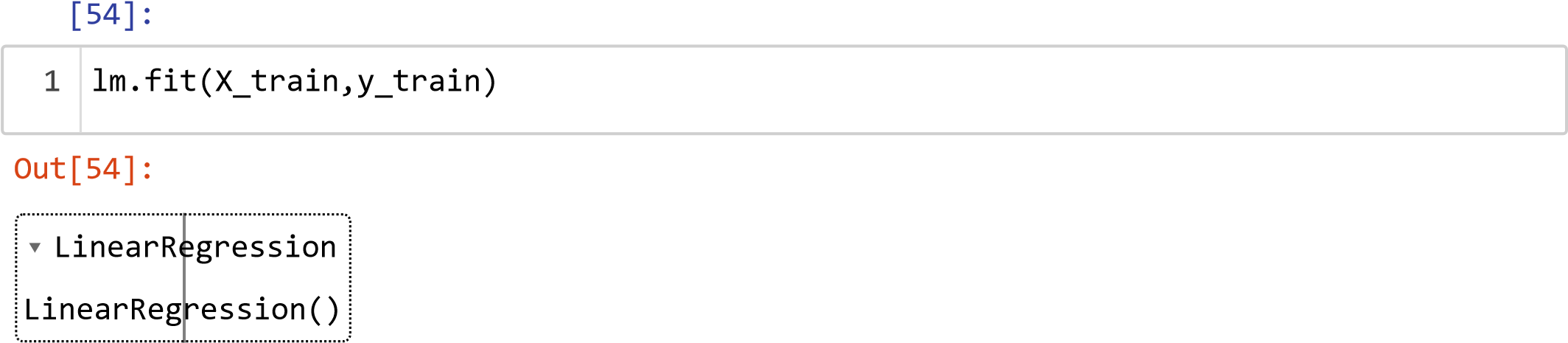
linear\_model

**import**

LinearRegression

1

In [53]:



In [55]:

predictions

**=**

lm

.

predict

(

X\_test

)

1

[58]:

sns

.

distplot

((

y\_test

**-**

predictions

)

,

bins

**=**

50

,

kde

**=**

**False**

)

;

1

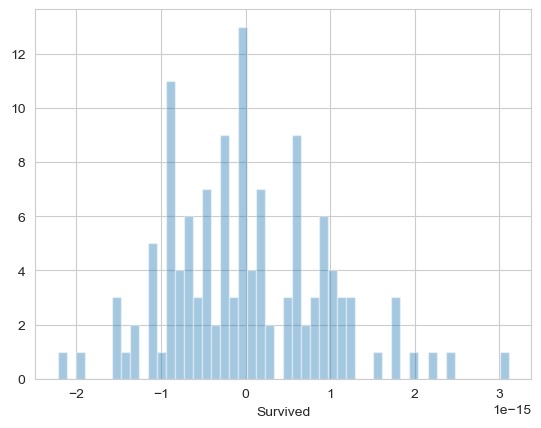
C:\Users\suraj\AppData\Local\Temp\ipykernel\_10448\3145007558.py:1: UserWa rning:

`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 w ith

similar flexibility) or `histplot` (an axes-level function for histogram s).

For a guide to updating your code to use the new functions, please see [https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (http s://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)](https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)  sns.distplot((y\_test-predictions),bins=50,kde=False);



In [59]:

**from**

sklearn

**import**

metrics

1

[60]:

1. print('MAE:', metrics.mean\_absolute\_error(y\_test, predictions))
2. print('MSE:', metrics.mean\_squared\_error(y\_test, predictions))
3. print('RMSE:', np.sqrt(metrics.mean\_squared\_error(y\_test, predictions)))

MAE: 7.128336721601204e-16

MSE: 8.392407869408556e-31

RMSE: 9.16100860681211e-16

In [ ]:

1