In [85]:
```
import pandas as pd
import pickle
```

In [86]:
```
data=pd.read_csv("/home/palacement/Downloads/Titanic Dataset.csv")
```

In [132]:
```
data.head()
```

Out[132]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [88]:
```
data.tail()
```

Out[88]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

In [89]:
```
data.isna().sum()
```

Out[89]:
```
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
```

In [90]:
```
data.head(10)
```

Out[90]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C

In [91]:
```
data.Pclass.unique()
```

Out[91]:
```
array([3, 1, 2])
```

In [92]:
```
data.Survived.unique()
```

Out[92]:
```
array([0, 1])
```

In [93]:
```
data.Age.unique()
```

Out[93]:
```
array([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.  ])
```

In [94]:
```
data.Sex.unique()
```

Out[94]:
```
array(['male', 'female'], dtype=object)
```

In [95]:
```
data.SibSp.unique()
```

Out[95]:
```
array([1, 0, 3, 4, 2, 5, 8])
```

In [96]:
```
data1=data.drop(['Age','PassengerId','Name','Cabin','SibSp','Parch','Ticket'],axis=1)
```

In [97]:
```
data1
```

Out[97]:

	Survived	Pclass	Sex	Fare	Embarked
0	0	3	male	7.2500	S
1	1	1	female	71.2833	C
2	1	3	female	7.9250	S
3	1	1	female	53.1000	S
4	0	3	male	8.0500	S
...	...	...	...	...	...
886	0	2	male	13.0000	S
887	1	1	female	30.0000	S
888	0	3	female	23.4500	S
889	1	1	male	30.0000	C
890	0	3	male	7.7500	Q

891 rows × 5 columns

In [98]:
```
data1['Sex']=data1['Sex'].map({'male':0,'female':1})
```

In [99]:
```
data1
```

Out[99]:

	Survived	Pclass	Sex	Fare	Embarked
0	0	3	0	7.2500	S
1	1	1	1	71.2833	C
2	1	3	1	7.9250	S
3	1	1	1	53.1000	S
4	0	3	0	8.0500	S
...	...	...	...	...	...
886	0	2	0	13.0000	S
887	1	1	1	30.0000	S
888	0	3	1	23.4500	S
889	1	1	0	30.0000	C
890	0	3	0	7.7500	Q

891 rows × 5 columns

In [ ]:
```
```

In [100]:
```
data2=data1.fillna(data.median())
```

```
/tmp/ipykernel_5462/1290514040.py:1: FutureWarning: The default value of numeric_only in DataFrame.median i
s 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.
  data2=data1.fillna(data.median())
```

In [ ]:
```
```

In [101]:
```
data2.isna().sum()
```

Out[101]:
```
Survived    0
Pclass      0
Sex         0
Fare        0
Embarked    2
dtype: int64
```

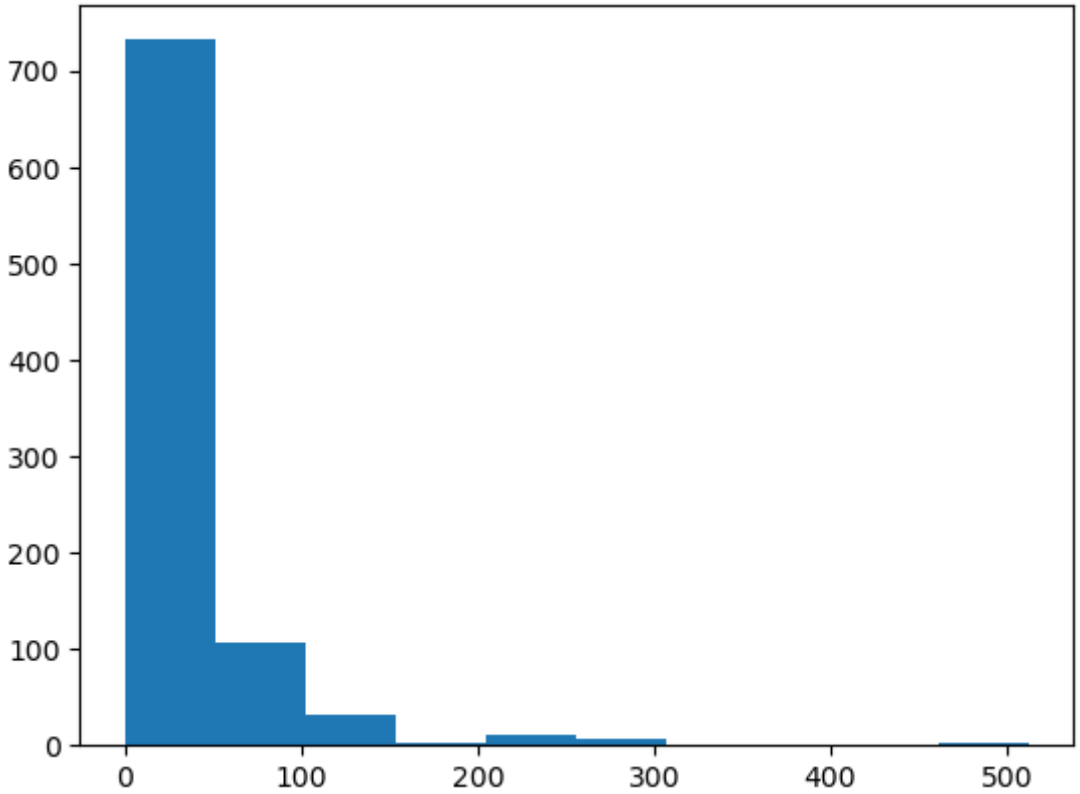
In [102]:
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.boxplot(data2,Age)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[102], line 3
      1 import seaborn as sns
      2 import matplotlib.pyplot as plt
----> 3 sns.boxplot(data2,Age)

NameError: name 'Age' is not defined
```

In [103]:
```
plt.hist(data2['Fare'])
```

Out[103]:
```
(array([732., 106.,  31.,   2.,  11.,   6.,   0.,   0.,   0.,   3.]),
 array([  0.     ,  51.23292, 102.46584, 153.69876, 204.93168, 256.1646 ,
        307.39752, 358.63044, 409.86336, 461.09628, 512.3292 ]),
 <BarContainer object of 10 artists>)
```



In [ ]:

In [104]:
```
data1.fillna(35,inplace=True)
```

In [ ]:

In [106]:
```
data1.describe
```

Out[106]:
```
<bound method NDFrame.describe of      Survived  Pclass  Sex     Fare Embarked
0           0       3    0   7.2500        S
1           1       1    1  71.2833        C
2           1       3    1   7.9250        S
3           1       1    1  53.1000        S
4           0       3    0   8.0500        S
..        ...     ...  ...      ...      ...
886         0       2    0  13.0000        S
887         1       1    1  30.0000        S
888         0       3    1  23.4500        S
889         1       1    0  30.0000        C
890         0       3    0   7.7500        Q

[891 rows x 5 columns]>
```

In [131]:
```
data1.groupby(['Fare'])
```

Out[131]:
```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7fa7ab85ee60>
```

In [108]:
```
data1['Pclass']=data1['Pclass'].map({1:'F',2:'S',3:'Third'})
```

In [109]:
```
data1.isna().sum()
```

Out[109]:
```
Survived    0
Pclass      0
Sex         0
Fare        0
Embarked    0
dtype: int64
```

In [110]:
```
data1=pd.get_dummies(data1)
```

In [111]:
```
data1.shape
```

Out[111]:
```
(891, 10)
```

In [112]:
```
data1.head(500)
```

Out[112]:

	Survived	Sex	Fare	Pclass_F	Pclass_S	Pclass_Third	Embarked_35	Embarked_C	Embarked_Q	Embarked_S
0	0	0	7.2500	0	0	1	0	0	0	1
1	1	1	71.2833	1	0	0	0	1	0	0
2	1	1	7.9250	0	0	1	0	0	0	1
3	1	1	53.1000	1	0	0	0	0	0	1
4	0	0	8.0500	0	0	1	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...
495	0	0	14.4583	0	0	1	0	1	0	0
496	1	1	78.2667	1	0	0	0	1	0	0
497	0	0	15.1000	0	0	1	0	0	0	1
498	0	1	151.5500	1	0	0	0	0	0	1
499	0	0	7.7958	0	0	1	0	0	0	1

500 rows × 10 columns

In [113]:
```
data1
```

Out[113]:

	Survived	Sex	Fare	Pclass_F	Pclass_S	Pclass_Third	Embarked_35	Embarked_C	Embarked_Q	Embarked_S
0	0	0	7.2500	0	0	1	0	0	0	1
1	1	1	71.2833	1	0	0	0	1	0	0
2	1	1	7.9250	0	0	1	0	0	0	1
3	1	1	53.1000	1	0	0	0	0	0	1
4	0	0	8.0500	0	0	1	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...
886	0	0	13.0000	0	1	0	0	0	0	1
887	1	1	30.0000	1	0	0	0	0	0	1
888	0	1	23.4500	0	0	1	0	0	0	1
889	1	0	30.0000	1	0	0	0	1	0	0
890	0	0	7.7500	0	0	1	0	0	1	0

891 rows × 10 columns

In [114]:
```
cor_mat=data1.corr()
cor_mat
```

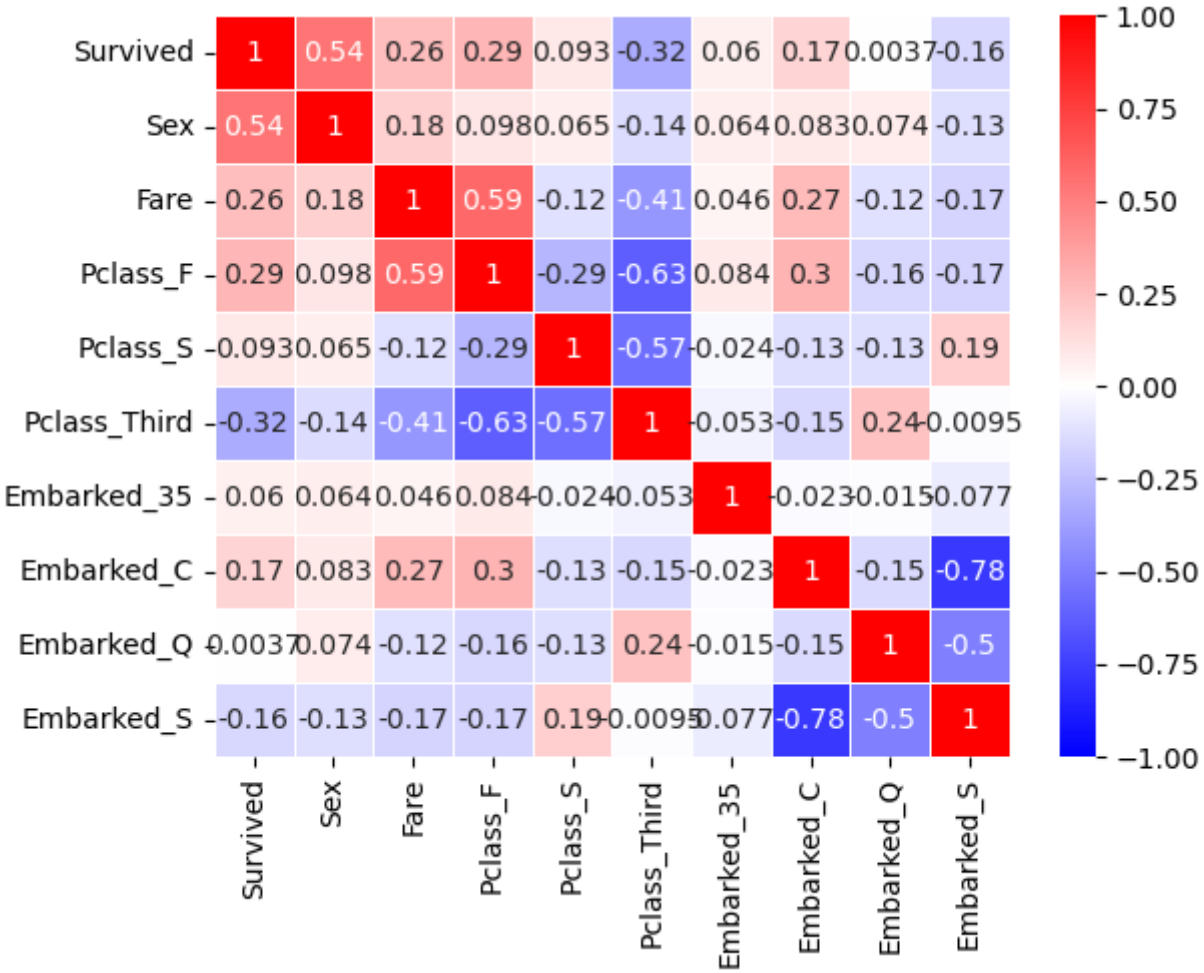
Out[114]:

	Survived	Sex	Fare	Pclass_F	Pclass_S	Pclass_Third	Embarked_35	Embarked_C	Embarked_Q	Embarked_S
Survived	1.000000	0.543351	0.257307	0.285904	0.093349	-0.322308	0.060095	0.168240	0.003650	-0.155660
Sex	0.543351	1.000000	0.182333	0.098013	0.064746	-0.137143	0.064296	0.082853	0.074115	-0.125722
Fare	0.257307	0.182333	1.000000	0.591711	-0.118557	-0.413333	0.045646	0.269335	-0.117216	-0.166603
Pclass_F	0.285904	0.098013	0.591711	1.000000	-0.288585	-0.626738	0.083847	0.296423	-0.155342	-0.170379
Pclass_S	0.093349	0.064746	-0.118557	-0.288585	1.000000	-0.565210	-0.024197	-0.125416	-0.127301	0.192061
Pclass_Third	-0.322308	-0.137143	-0.413333	-0.626738	-0.565210	1.000000	-0.052550	-0.153329	0.237449	-0.009511
Embarked_35	0.060095	0.064296	0.045646	0.083847	-0.024197	-0.052550	1.000000	-0.022864	-0.014588	-0.076588
Embarked_C	0.168240	0.082853	0.269335	0.296423	-0.125416	-0.153329	-0.022864	1.000000	-0.148258	-0.778359
Embarked_Q	0.003650	0.074115	-0.117216	-0.155342	-0.127301	0.237449	-0.014588	-0.148258	1.000000	-0.496624
Embarked_S	-0.155660	-0.125722	-0.166603	-0.170379	0.192061	-0.009511	-0.076588	-0.778359	-0.496624	1.000000

In [115]:
```
import seaborn as sns
```

In [116]:
```
sns.heatmap(cor_mat,vmax=1,vmin=-1,annot=True,linewidth=.5,cmap='bwr')
```

Out[116]: <Axes: >



In [117]:
```
data.groupby('Survived').count()
```

Out[117]:

Survived	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	549	549	549	549	424	549	549	549	549	68	549
1	342	342	342	342	290	342	342	342	342	136	340

In [118]:
```
y=data1['Survived']
x=data1.drop('Survived',axis=1)
```

In [153]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
```

In [154]:
```
x_test.head(5)
```

Out[154]:

	Sex	Fare	Pclass_F	Pclass_S	Pclass_Third	Embarked_35	Embarked_C	Embarked_Q	Embarked_S
709	0	15.2458	0	0	1	0	1	0	0
439	0	10.5000	0	1	0	0	0	0	1
840	0	7.9250	0	0	1	0	0	0	1
720	1	33.0000	0	1	0	0	0	0	1
39	1	11.2417	0	0	1	0	1	0	0

In [158]:
```
from sklearn.linear_model import LogisticRegression
classifier=LogisticRegression()
classifier.fit (x_train,y_train)
```

Out[158]:
```
▾ LogisticRegression

LogisticRegression()
```

In [159]:
```
y_pred=classifier.predict(x_test)
```

In [160]:
```
y_pred
```

Out[160]:
```
array([0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1,
       0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
       1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0,
       0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0,
       1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0,
       1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0,
       0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1,
       0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 1, 0])
```

In [167]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

Out[167]:
```
array([[144,  31],
       [ 34,  86]])
```

In [163]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
```

Out[163]:
```
0.08690476190476193
```

In [164]:
```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_pred,y_test)
```

Out[164]:
```
0.22033898305084745
```

In [ ]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred_ridge)
```

In [168]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Out[168]:
```
0.7796610169491526
```

In [169]:
```
from sklearn.metrics import classification_report
classification_report(y_test,y_pred)
```

Out[169]:
```
'              precision    recall  f1-score   support\n\n           0       0.81      0.82      0.82
175\n           1       0.74      0.72      0.73       120\n\n    accuracy                           0.78
295\n   macro avg       0.77      0.77      0.77       295\nweighted avg       0.78      0.78      0.78
295\n'
```

In [170]:
```
from sklearn.pipeline import Pipeline
Pipeline(y_test,y_pred)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[170], line 2
      1 from sklearn.pipeline import Pipeline
----> 2 Pipeline(y_test,y_pred)

TypeError: Pipeline.__init__() takes 2 positional arguments but 3 were given
```

In [171]:
```
from sklearn.model_selection import cross_val_score
cross_val_score(y_test,y_pred)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[171], line 2
      1 from sklearn.model_selection import cross_val_score
----> 2 cross_val_score(y_test,y_pred)

File ~/anaconda3/lib/python3.10/site-packages/sklearn/model_selection/_validation.py:513, in cross_val_score(estimator, X, y, groups, scoring, cv, n_jobs, verbose, fit_params, pre_dispatch, error_score)
    395 """Evaluate a score by cross-validation.
    396
    397 Read more in the :ref:`User Guide <cross_validation>`.
   (...)
    510 [0.3315057  0.08022103 0.03531816]
    511 """
    512 # To ensure multimetric format is not supported
--> 513 scorer = check_scoring(estimator, scoring=scoring)
    515 cv_results = cross_validate(
    516     estimator=estimator,
    517     X=X,
   (...)
    526     error_score=error_score,
    527 )
    528 return cv_results["test_score"]

File ~/anaconda3/lib/python3.10/site-packages/sklearn/metrics/_scorer.py:474, in check_scoring(estimator, scoring, allow_none)
    448 """Determine scorer from user options.
    449
    450 A TypeError will be thrown if the estimator cannot be scored.
   (...)
    471     ``scorer(estimator, X, y)``.
    472 """
    473 if not hasattr(estimator, "fit"):
--> 474     raise TypeError(
    475         "estimator should be an estimator implementing 'fit' method, %r was passed"
    476         % estimator
    477     )
    478 if isinstance(scoring, str):
```

```
    479     return get_scorer(scoring)

TypeError: estimator should be an estimator implementing 'fit' method, 709    1
439    0
840    0
720    1
39     1
      ..
715    0
525    0
381    1
140    0
173    0
Name: Survived, Length: 295, dtype: int64 was passed
```

In [ ]: