In [81]:

```
import pandas as pd
import pickle
import warnings
warnings.filterwarnings("ignore")
```

In [82]:

```
data=pd.read_csv("C:\\Users\\reshma_koduri\\OneDrive\\Documents\\Titanic Dataset crt
data
```

Out[82]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN
...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN

891 rows × 12 columns

In [83]:
```
data.describe()
```

Out[83]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [84]:
```
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   Sex          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 [85]:
```
data.isna().sum()
```

Out[85]:
```
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 [ ]:

In [86]:

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

Out[86]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	male	22.0	7.2500	S
1	1	1	female	38.0	71.2833	C
2	1	3	female	26.0	7.9250	S
3	1	1	female	35.0	53.1000	S
4	0	3	male	35.0	8.0500	S
...	...	...	...	...	...	...
886	0	2	male	27.0	13.0000	S
887	1	1	female	19.0	30.0000	S
888	0	3	female	NaN	23.4500	S
889	1	1	male	26.0	30.0000	C
890	0	3	male	32.0	7.7500	Q

891 rows × 6 columns

In [87]:

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

In [88]:

```
data2=pd.get_dummies(data1,dtype=int)
data2
```

Out[88]:

	Survived	Pclass	Sex	Age	Fare	Embarked_C	Embarked_Q	Embarked_S
0	0	3	1	22.0	7.2500	0	0	1
1	1	1	0	38.0	71.2833	1	0	0
2	1	3	0	26.0	7.9250	0	0	1
3	1	1	0	35.0	53.1000	0	0	1
4	0	3	1	35.0	8.0500	0	0	1
...	...	...	...	...	...	...	...	...
886	0	2	1	27.0	13.0000	0	0	1
887	1	1	0	19.0	30.0000	0	0	1
888	0	3	0	NaN	23.4500	0	0	1
889	1	1	1	26.0	30.0000	1	0	0
890	0	3	1	32.0	7.7500	0	1	0

891 rows × 8 columns

In [89]:

```
data2['Age']=data2['Age'].mask(data2['Age']>65,60) # removing boundry values
```

In [90]:

```
colnames=list(data2)
colnames
```

Out[90]:

```
['Survived',
 'Pclass',
 'Sex',
 'Age',
 'Fare',
 'Embarked_C',
 'Embarked_Q',
 'Embarked_S']
```

In [91]:

```
from sklearn.impute import KNNImputer
imputer=KNNImputer(n_neighbors=3)
data_filled=imputer.fit_transform(data2)
data2=pd.DataFrame(data=data_filled,columns=colnames)
```

In [92]:

```
data2['Age'].unique()
```

Out[92]:

```
array([22.        , 38.        , 26.        , 35.        , 55.16666667,
       54.        ,  2.        , 27.        , 14.        ,  4.        ,
       58.        , 20.        , 39.        , 55.        , 35.66666667,
       31.        , 16.66666667, 34.        , 15.        , 28.        ,
        8.        , 38.5       , 19.        , 40.        , 26.97333333,
       18.        , 60.        , 42.        , 23.66666667, 21.        ,
       32.16666667,  3.        , 25.33333333, 36.        , 18.66666667,
        7.        , 49.        , 29.        , 65.        , 43.        ,
       28.5       ,  5.        , 11.        , 45.        , 33.        ,
       17.        , 32.        , 16.        , 25.        ,  0.83      ,
       30.        , 23.        , 24.        , 46.        , 59.        ,
       37.        , 24.33333333, 22.66666667, 47.        , 14.5       ,
       32.5       , 12.        , 14.66666667,  9.        , 36.5       ,
       51.        , 55.5       , 40.5       , 34.33333333, 28.33333333,
       44.        ,  1.        , 57.66666667, 61.        , 56.        ,
       50.        , 48.33333333, 45.5       , 20.5       , 33.33333333,
       29.33333333, 25.83333333, 62.        , 41.        , 55.33333333,
       52.        , 37.16666667, 45.33333333, 63.        , 31.66666667,
       23.5       , 46.33333333, 38.33333333,  0.92      , 43.66666667,
       20.33333333, 39.66666667, 35.33333333, 21.66666667, 10.        ,
       64.        , 26.33333333, 13.        , 22.33333333, 48.        ,
        0.75      , 23.33333333, 31.83333333, 23.16666667, 42.33333333,
       24.66666667, 32.66666667, 31.16666667, 28.66666667, 34.5       ,
       53.        , 16.5       , 33.66666667, 57.        , 28.83333333,
       24.5       , 22.16666667,  6.        ,  0.67      , 30.5       ,
       50.33333333,  0.42      , 38.66666667, 21.33333333])
```

In [93]:

```
data2
```

Out[93]:

	Survived	Pclass	Sex	Age	Fare	Embarked_C	Embarked_Q	Embarked_S
0	0.0	3.0	1.0	22.000000	7.2500	0.0	0.0	1.0
1	1.0	1.0	0.0	38.000000	71.2833	1.0	0.0	0.0
2	1.0	3.0	0.0	26.000000	7.9250	0.0	0.0	1.0
3	1.0	1.0	0.0	35.000000	53.1000	0.0	0.0	1.0

	Survived	Pclass	Sex	Age	Fare	Embarked_C	Embarked_Q	Embarked_S
4	0.0	3.0	1.0	35.000000	8.0500	0.0	0.0	1.0
...	...	...	...	...	...	...	...	...
886	0.0	2.0	1.0	27.000000	13.0000	0.0	0.0	1.0
887	1.0	1.0	0.0	19.000000	30.0000	0.0	0.0	1.0
888	0.0	3.0	0.0	21.333333	23.4500	0.0	0.0	1.0
889	1.0	1.0	1.0	26.000000	30.0000	1.0	0.0	0.0
890	0.0	3.0	1.0	32.000000	7.7500	0.0	1.0	0.0

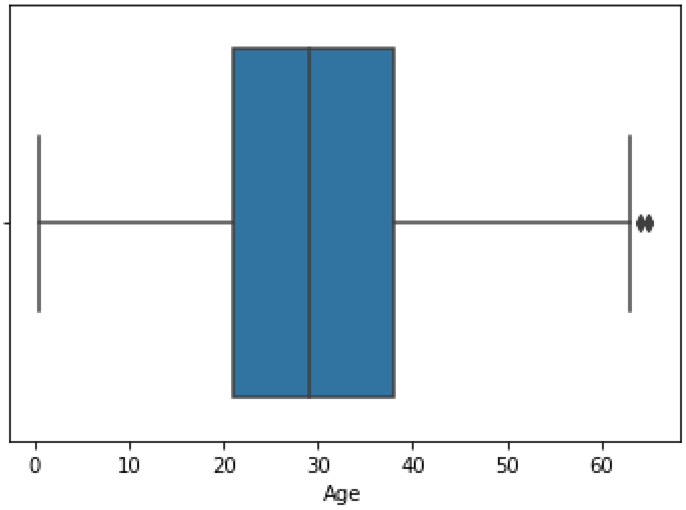
891 rows × 8 columns

In [94]:

```
import seaborn as sb
import matplotlib.pyplot as plt
sb.boxplot(data2.Age)
```

Out[94]:

```
<AxesSubplot:xlabel='Age'>
```

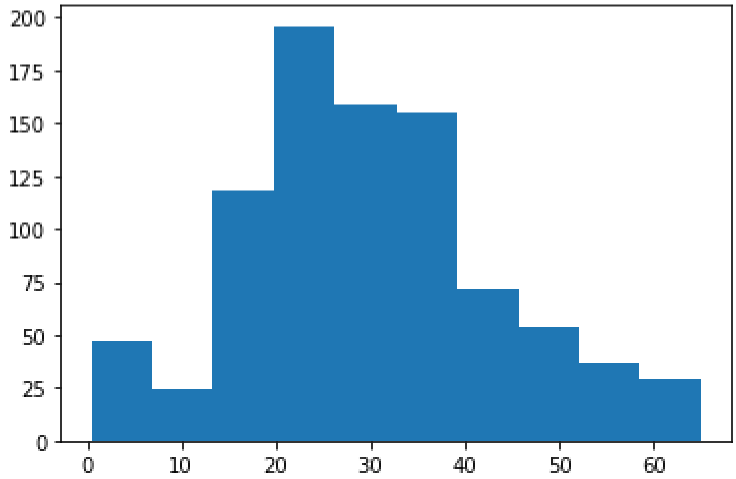


In [95]:

```
import matplotlib.pyplot as plt
plt.hist(data2['Age'])
```

Out[95]:

```
(array([ 47.,  24., 118., 196., 159., 155.,  72.,  54.,  37.,  29.]),
 array([ 0.42 ,  6.878, 13.336, 19.794, 26.252, 32.71 , 39.168, 45.626,
        52.084, 58.542, 65.   ]),
 <BarContainer object of 10 artists>)
```



In [96]:

```
cor_mat=data2.corr()
cor_mat
```

Out[96]:

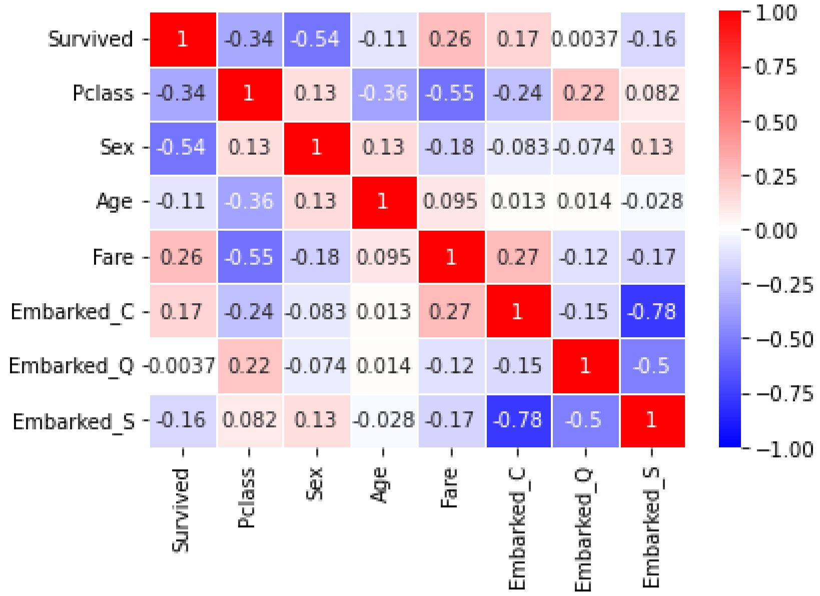
	Survived	Pclass	Sex	Age	Fare	Embarked_C	Embarked_Q	Embarl
Survived	1.000000	-0.338481	-0.543351	-0.109370	0.257307	0.168240	0.003650	-0.15
Pclass	-0.338481	1.000000	0.131900	-0.359539	-0.549500	-0.243292	0.221009	0.08
Sex	-0.543351	0.131900	1.000000	0.132870	-0.182333	-0.082853	-0.074115	0.12
Age	-0.109370	-0.359539	0.132870	1.000000	0.095431	0.013323	0.013657	-0.02
Fare	0.257307	-0.549500	-0.182333	0.095431	1.000000	0.269335	-0.117216	-0.16
Embarked_C	0.168240	-0.243292	-0.082853	0.013323	0.269335	1.000000	-0.148258	-0.77
Embarked_Q	0.003650	0.221009	-0.074115	0.013657	-0.117216	-0.148258	1.000000	-0.49
Embarked_S	-0.155660	0.081720	0.125722	-0.027604	-0.166603	-0.778359	-0.496624	1.00

In [97]:

```
import seaborn as sb
sb.heatmap(cor_mat,vmax=1,vmin=-1,annot=True,linewidth=.5,cmap="bwr")
```

Out[97]:

```
<AxesSubplot:>
```



In [98]:

```
y=data2['Survived']
x=data2.drop(['Survived'],axis=1)
```

In [99]:

```
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 [100…

```
from sklearn.linear_model import LogisticRegression
classifier=LogisticRegression()
classifier.fit(x_train, y_train)
```

Out[100…

```
LogisticRegression()
```

In [101…

```
ypred=classifier.predict(x_test)
ypred
```

Out[101…

```
array([0., 0., 0., 1., 1., 1., 1., 0., 1., 1., 0., 0., 0., 0., 0., 1., 0.,
       1., 0., 0., 0., 0., 1., 0., 0., 0., 1., 0., 0., 0., 0., 1., 0., 0.,
       0., 0., 0., 1., 1., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 1.,
       1., 0., 1., 0., 1., 0., 1., 1., 1., 0., 1., 1., 0., 0., 1., 0., 0.,
       0., 1., 1., 1., 1., 1., 0., 0., 1., 1., 1., 0., 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., 1., 1., 0., 1., 0., 0., 0., 0., 0., 1.,
       0., 0., 1., 1., 0., 0., 1., 1., 1., 1., 0., 1., 0., 0., 1., 0., 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., 0., 0., 1., 0., 0.,
       1., 1., 0., 1., 0., 0., 0., 1., 1., 0., 0., 0., 0., 1., 1., 0., 0.,
       0., 0., 1., 0., 0., 0., 0., 1., 1., 1., 0., 1., 1., 0., 0., 1., 0.,
       0., 1., 0., 0., 0., 0., 1., 0., 1., 0., 0., 0., 1., 0., 1., 0., 0.,
       1., 0., 0., 0., 1., 0., 1., 1., 1., 0., 1., 0., 1., 0., 1., 1., 1.,
       1., 0., 0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 1., 0., 1., 1., 0., 1., 0., 0., 0., 0., 0., 1., 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 [102…

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,ypred)
```

Out[102…

```
array([[154,  21],
       [ 32,  88]], dtype=int64)
```

In [103…

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,ypred)
```

Out[103…

```
0.8203389830508474
```

In [ ]: