In [1]:

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

In [2]:

```
data=pd.read_csv("/home/placement/Downloads/Titanic Dataset.csv")
```

In [3]:

```
data.describe()
```

Out[3]:

	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 [4]:

```
data
```

Out[4]:

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

891 rows × 12 columns

In [5]:

```
data['Pclass'].unique()
```

Out[5]:

```
array([3, 1, 2])
```

In [6]:

```
data['Survived'].unique()
```

Out[6]:

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

In [7]:

```
data['SibSp'].unique()
```

Out[7]:

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

In [8]:

```
data['Parch'].unique()
```

Out[8]:

```
array([0, 1, 2, 5, 3, 4, 6])
```

In [9]:

```
list(data)
```

Out[9]:

```
['PassengerId',
 'Survived',
 'Pclass',
 'Name',
 'Sex',
 'Age',
 'SibSp',
 'Parch',
 'Ticket',
 'Fare',
 'Cabin',
 'Embarked']
```

In [ ]:

```

```

In [10]:

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

In [11]:

```
data1
```

Out[11]:

	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 [12]:

```
data1.isna().sum()
```

Out[12]:

```
Survived      0
Pclass        0
Sex           0
Age         177
Fare          0
Embarked      2
dtype: int64
```

data1=pd.get_dummies(data1)

In [13]:

```
data1
```

Out[13]:

```
     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 [14]:

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

In [15]:

```
data2=data1.fillna(data1.median())
```

In [16]:

```
data2
```

Out[16]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	1	22.0	7.2500	S
1	1	1	0	38.0	71.2833	C
2	1	3	0	26.0	7.9250	S
3	1	1	0	35.0	53.1000	S
4	0	3	1	35.0	8.0500	S
...	...	...	...	...	...	...
886	0	2	1	27.0	13.0000	S
887	1	1	0	19.0	30.0000	S
888	0	3	0	28.0	23.4500	S
889	1	1	1	26.0	30.0000	C
890	0	3	1	32.0	7.7500	Q

891 rows × 6 columns

In [17]:

```
data2.isna().sum()
```

Out[17]:

```
Survived    0
Pclass      0
Sex         0
Age         0
Fare        0
Embarked    2
dtype: int64
```

In [18]:

```
data2['Pclass']=data2['Pclass'].map({1:'F',2:'S',3:'third'})
```

In [19]:

```
data2
```

Out[19]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	third	1	22.0	7.2500	S
1	1	F	0	38.0	71.2833	C
2	1	third	0	26.0	7.9250	S
3	1	F	0	35.0	53.1000	S
4	0	third	1	35.0	8.0500	S
...	...	...	...	...	...	...
886	0	S	1	27.0	13.0000	S
887	1	F	0	19.0	30.0000	S
888	0	third	0	28.0	23.4500	S
889	1	F	1	26.0	30.0000	C
890	0	third	1	32.0	7.7500	Q

891 rows × 6 columns

In [20]:

```
data2=pd.get_dummies(data2)
```

In [21]:

```
data2
```

Out[21]:

	Survived	Sex	Age	Fare	Pclass_F	Pclass_S	Pclass_third	Embarked_C	Embarked_(
0	0	1	22.0	7.2500	0	0	1	0	
1	1	0	38.0	71.2833	1	0	0	1	
2	1	0	26.0	7.9250	0	0	1	0	
3	1	0	35.0	53.1000	1	0	0	0	
4	0	1	35.0	8.0500	0	0	1	0	
...	...	...	...	...	...	...	...	...	.
886	0	1	27.0	13.0000	0	1	0	0	
887	1	0	19.0	30.0000	1	0	0	0	
888	0	0	28.0	23.4500	0	0	1	0	
889	1	1	26.0	30.0000	1	0	0	1	
890	0	1	32.0	7.7500	0	0	1	0	

891 rows × 10 columns

In [22]:

```
data2.isna().sum()
```

Out[22]:

```
Survived        0
Sex             0
Age             0
Fare            0
Pclass_F        0
Pclass_S        0
Pclass_third    0
Embarked_C      0
Embarked_Q      0
Embarked_S      0
dtype: int64
```

In [23]:

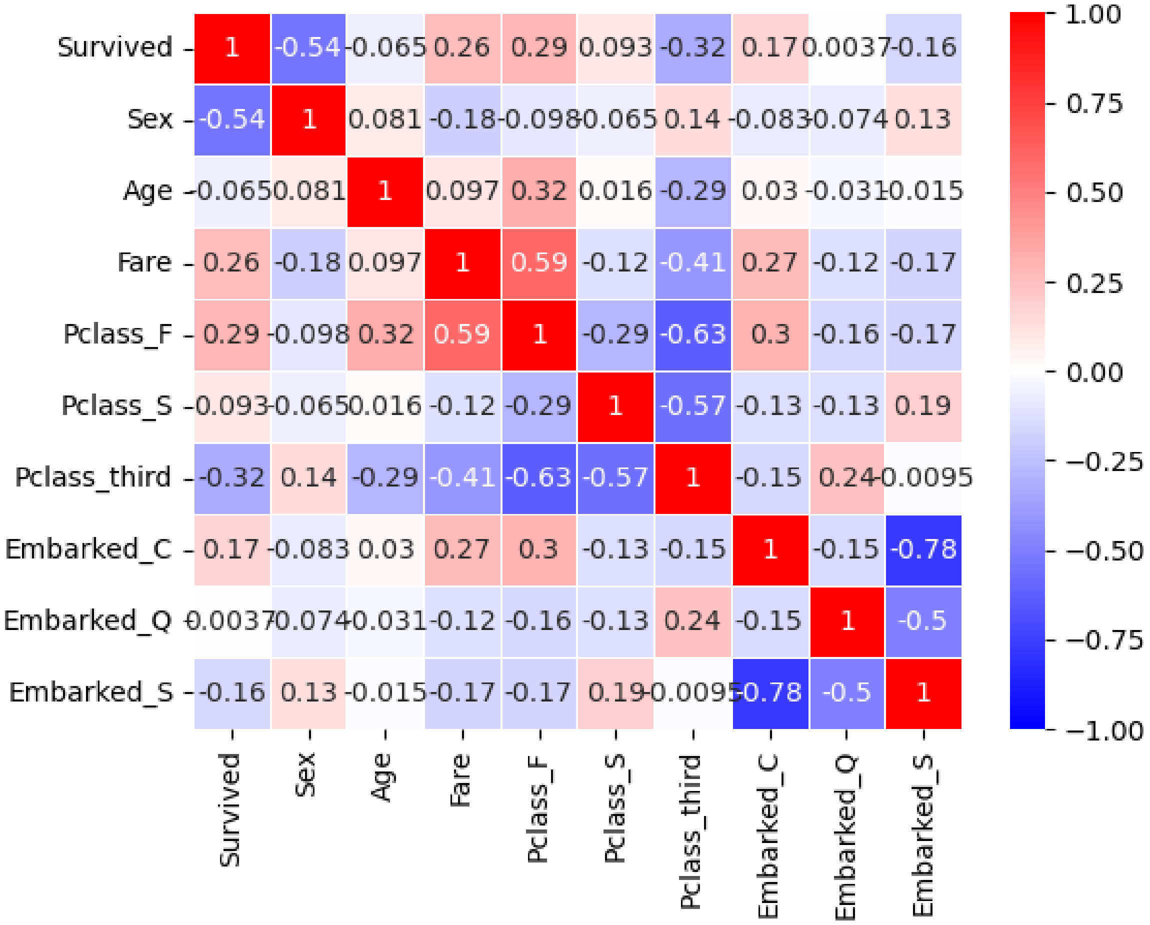
```
cor=data2.corr()
```

In [24]:

```
import seaborn as sns
sns.heatmap(cor,vmax=1,vmin=-1,annot=True,linewidth=.5,cmap='bwr')
```

Out[24]:

```
<Axes: >
```

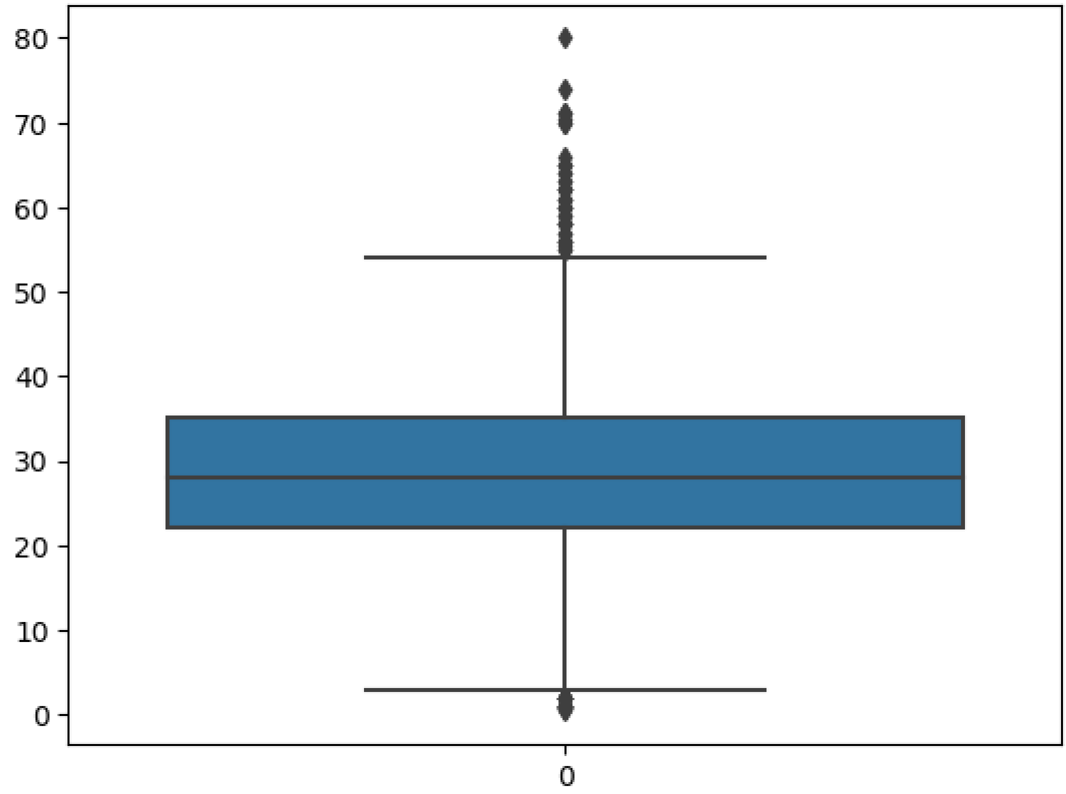


In [25]:

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

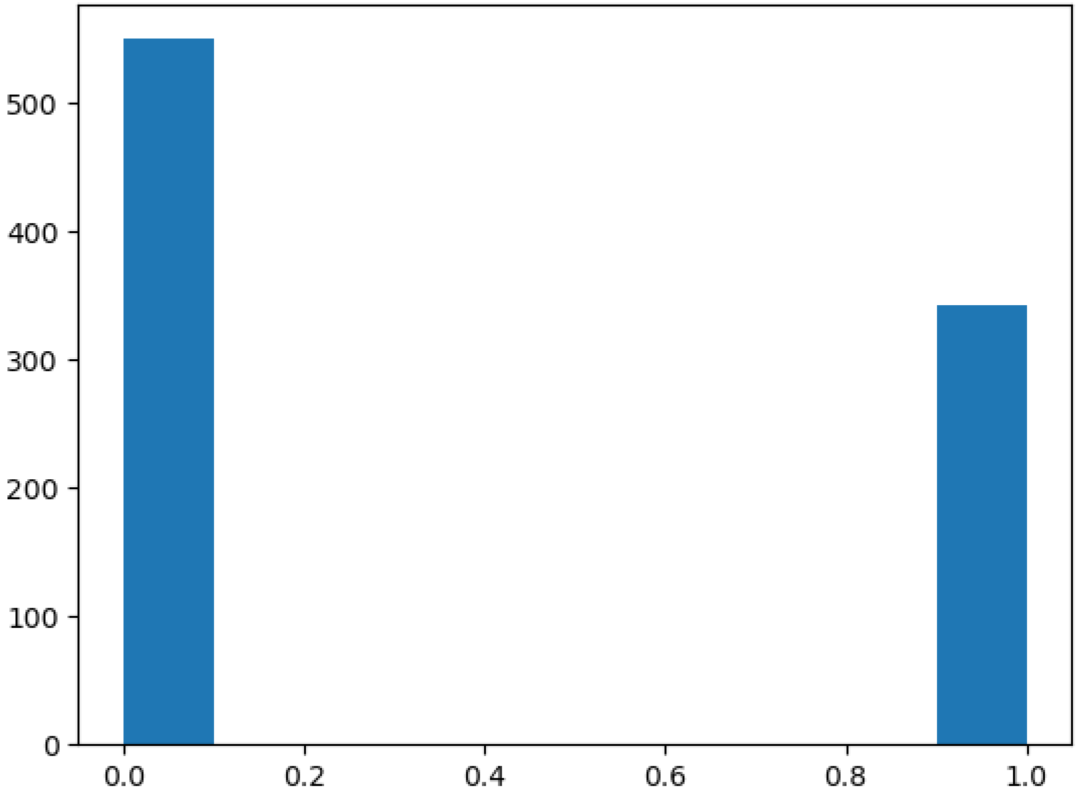
Out[25]:

```
<Axes: >
```



In [26]:

```
plt.hist(data2['Survived'])
plt.show()
```

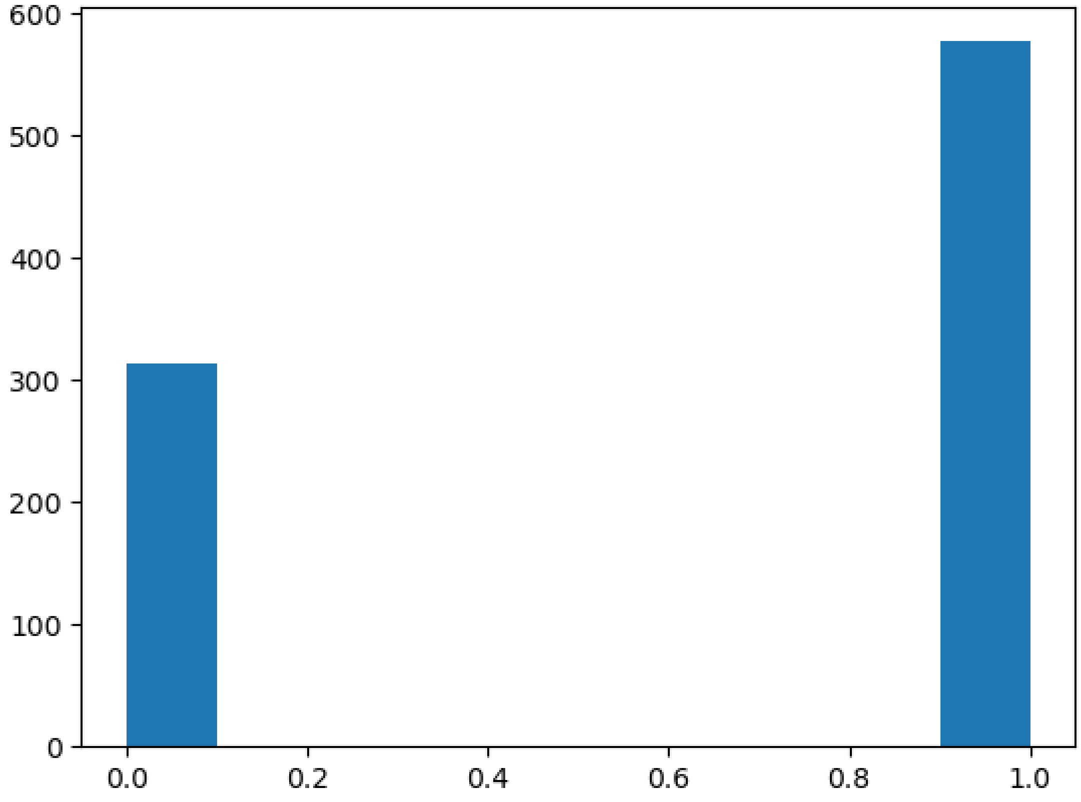


In [27]:

```
plt.hist(data2['Sex'])
```

Out[27]:

```
(array([314.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0., 577.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <BarContainer object of 10 artists>)
```

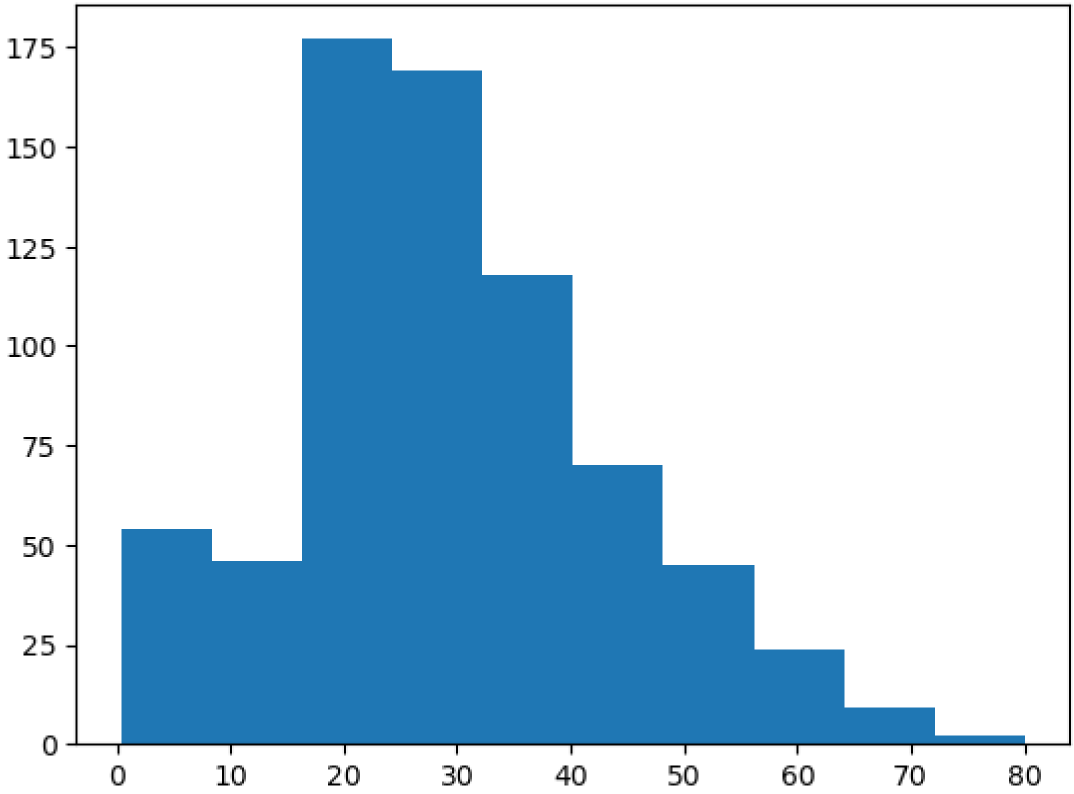


In [28]:

```
plt.hist(data1['Age'])
```

Out[28]:

```
(array([ 54.,  46., 177., 169., 118.,  70.,  45.,  24.,   9.,   2.]),
 array([ 0.42 ,  8.378, 16.336, 24.294, 32.252, 40.21 , 48.168, 56.12
6,
        64.084, 72.042, 80.   ]),
 <BarContainer object of 10 artists>)
```

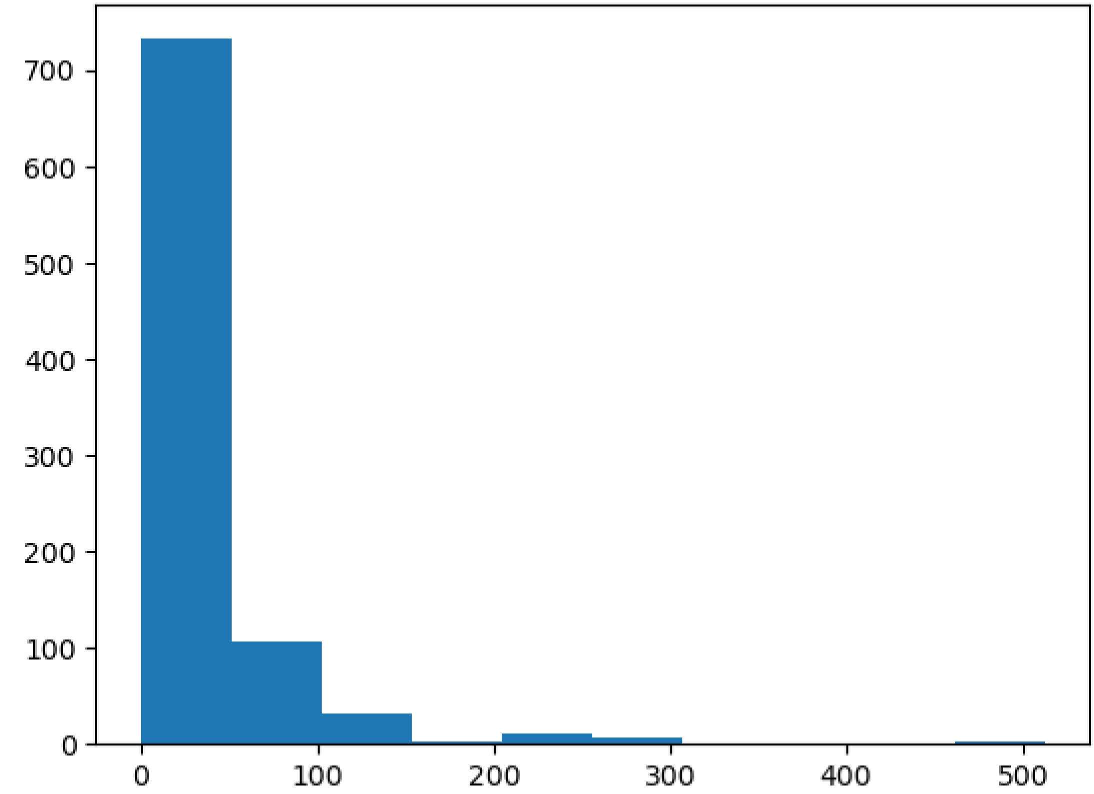


In [29]:

```
plt.hist(data1['Fare'])
```

Out[29]:

```
(array([732., 106.,  31.,   2.,  11.,   6.,   0.,   0.,   0.,   3.]),
 array([  0.     ,  51.23292, 102.46584, 153.69876, 204.93168, 256.164
6 ,
        307.39752, 358.63044, 409.86336, 461.09628, 512.3292 ]),
 <BarContainer object of 10 artists>)
```



In [ ]:

```

```

In [30]:

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

In [31]:

```
y
```

Out[31]:

```
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 [32]:

```
x
```

Out[32]:

	Sex	Age	Fare	Pclass_F	Pclass_S	Pclass_third	Embarked_C	Embarked_Q	Embark
0	1	22.0	7.2500	0	0	1	0	0	
1	0	38.0	71.2833	1	0	0	1	0	
2	0	26.0	7.9250	0	0	1	0	0	
3	0	35.0	53.1000	1	0	0	0	0	
4	1	35.0	8.0500	0	0	1	0	0	
...	...	...	...	...	...	...	...	...	
886	1	27.0	13.0000	0	1	0	0	0	
887	0	19.0	30.0000	1	0	0	0	0	
888	0	28.0	23.4500	0	0	1	0	0	
889	1	26.0	30.0000	1	0	0	1	0	
890	1	32.0	7.7500	0	0	1	0	1	

891 rows × 9 columns

In [ ]:

In [33]:

```
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 [34]:

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

Out[34]:

```
▾ LogisticRegression

LogisticRegression()
```

In [35]:

```
y_pred=clas.predict(x_test)
```

In [36]:

```
y_pred
```

Out[36]:

```
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, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0,
0,
       1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 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,
0,
       1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 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, 0, 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, 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 [37]:

```
x_test
```

Out[37]:

	Sex	Age	Fare	Pclass_F	Pclass_S	Pclass_third	Embarked_C	Embarked_Q	Embark
709	1	28.0	15.2458	0	0	1	1	0	
439	1	31.0	10.5000	0	1	0	0	0	
840	1	20.0	7.9250	0	0	1	0	0	
720	0	6.0	33.0000	0	1	0	0	0	
39	0	14.0	11.2417	0	0	1	1	0	
...	...	...	...	...	...	...	...	...	
715	1	19.0	7.6500	0	0	1	0	0	
525	1	40.5	7.7500	0	0	1	0	1	
381	0	1.0	15.7417	0	0	1	1	0	
140	0	28.0	15.2458	0	0	1	1	0	
173	1	21.0	7.9250	0	0	1	0	0	

295 rows × 9 columns

In [ ]:

In [38]:

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

Out[38]:

```
array([[154,  21],
       [ 37,  83]])
```

In [39]:

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

Out[39]:

```
0.8033898305084746
```

In [ ]:

In [ ]: