In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
train=pd.read_csv('D:/archive/train.csv')
train
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	Na
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C8
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	Na
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C12
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	Na
...	...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	Na
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B4
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	Na
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C14
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	Na

891 rows × 12 columns

In [3]:
```
train.head()
```

Out[3]:

	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

EXPLORATORY DATA ANALYSIS

missing data

In [4]:
```
train.isnull()
```

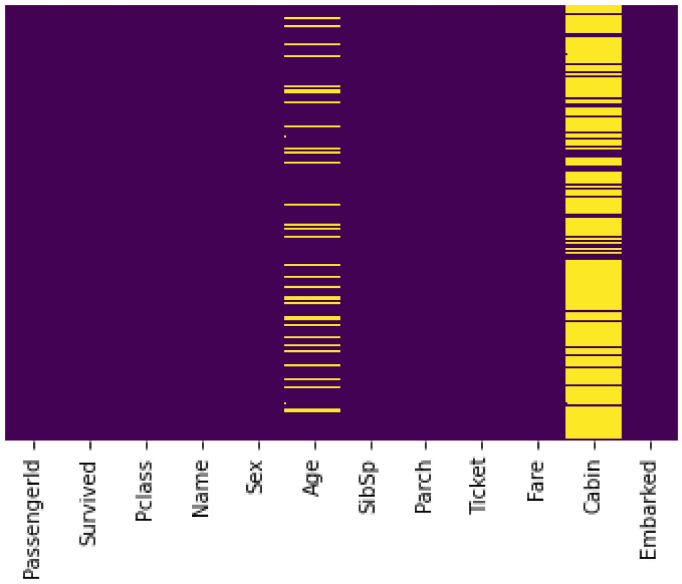
Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embark
0	False	False	False	False	False	False	False	False	False	False	True	Fal
1	False	False	False	False	False	False	False	False	False	False	False	Fal
2	False	False	False	False	False	False	False	False	False	False	True	Fal
3	False	False	False	False	False	False	False	False	False	False	False	Fal
4	False	False	False	False	False	False	False	False	False	False	True	Fal
...	...	...	...	...	...	...	...	...	...	...	...	
886	False	False	False	False	False	False	False	False	False	False	True	Fal
887	False	False	False	False	False	False	False	False	False	False	False	Fal
888	False	False	False	False	False	True	False	False	False	False	True	Fal
889	False	False	False	False	False	False	False	False	False	False	False	Fal
890	False	False	False	False	False	False	False	False	False	False	True	Fal

891 rows × 12 columns

In [5]:
```
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

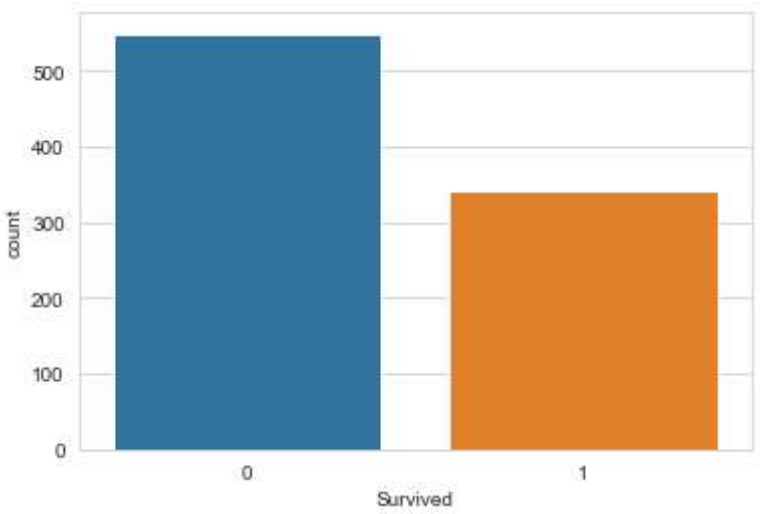
Out[5]:
```
<AxesSubplot:>
```



roughly 20% of age data is missing from the dataset.The proportion of missing age data is likely small enough than the cabin missing data which is roughly more than 95%.

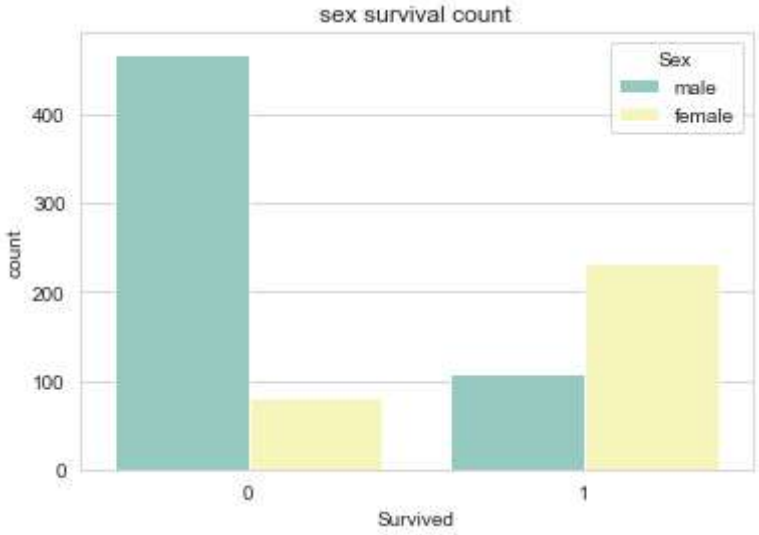
In [6]:
```
sns.set_style("whitegrid")
sns.countplot(x='Survived',data=train)
```

Out[6]:
```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```



In [7]:
```
plt.title("sex survival count")
sns.set_style("whitegrid")
sns.countplot(x='Survived',hue='Sex',data=train,palette="Set3")
```

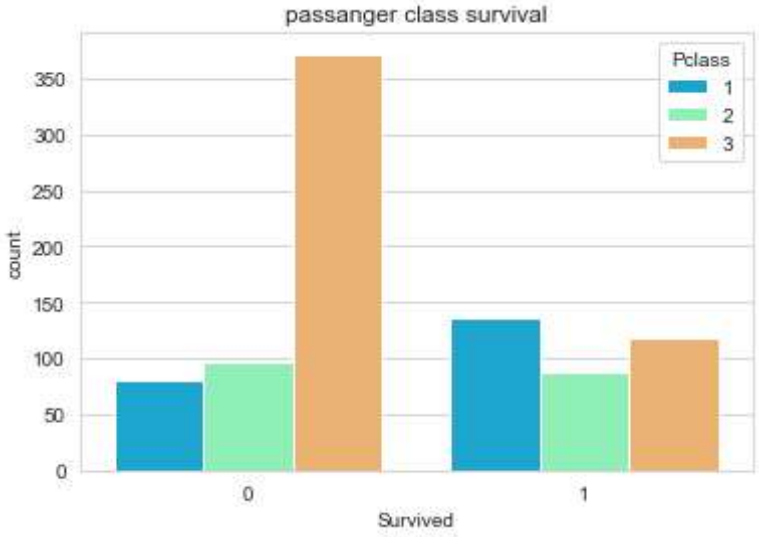
Out[7]:
```
<AxesSubplot:title={'center':'sex survival count'}, xlabel='Survived', ylabel='coun
t'>
```



This graph interperates that the count of men deaths are far more than women deaths, hecne the count of women survival are more than men

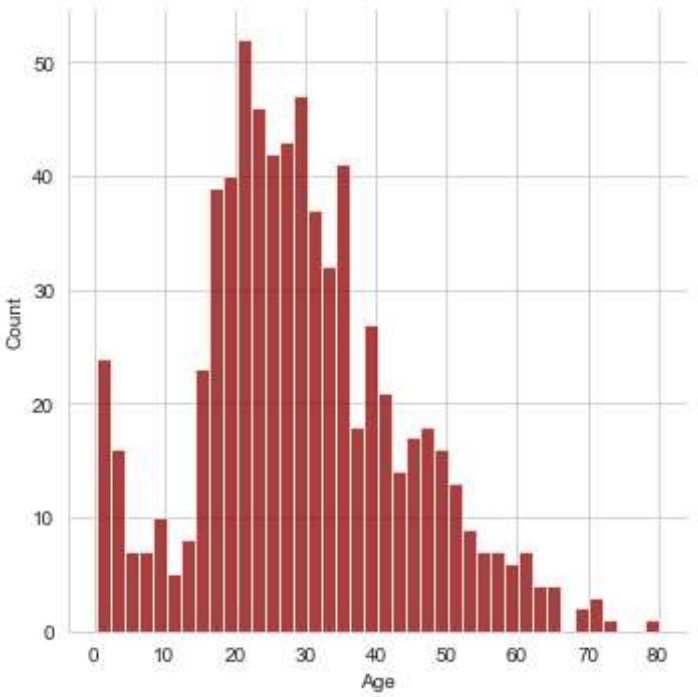
In [8]:
```
plt.title("passanger class survival")
sns.set_style("whitegrid")
sns.countplot(x='Survived',hue='Pclass',data=train,palette="rainbow")
```

Out[8]:
```
<AxesSubplot:title={'center':'passanger class survival'}, xlabel='Survived', ylabel
='count'>
```



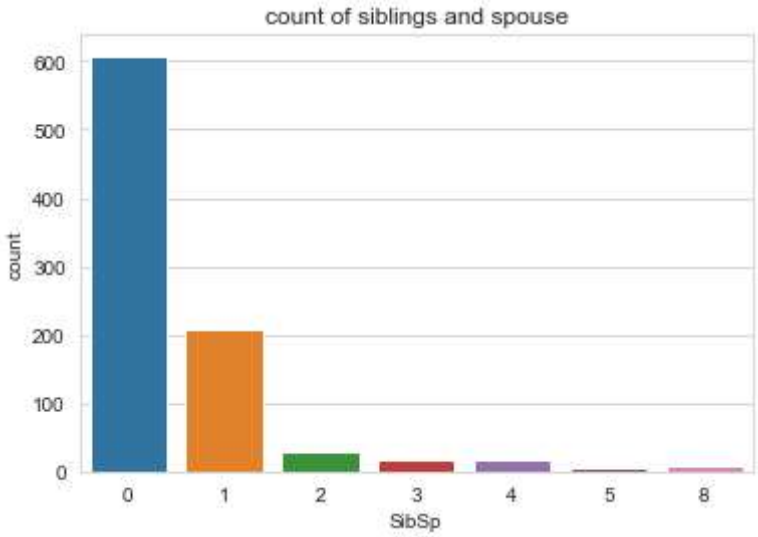
In [9]:
```
sns.displot(train['Age'].dropna(),kde=False,color='darkred',bins=40)
```

Out[9]:
```
<seaborn.axisgrid.FacetGrid at 0x264906e2760>
```



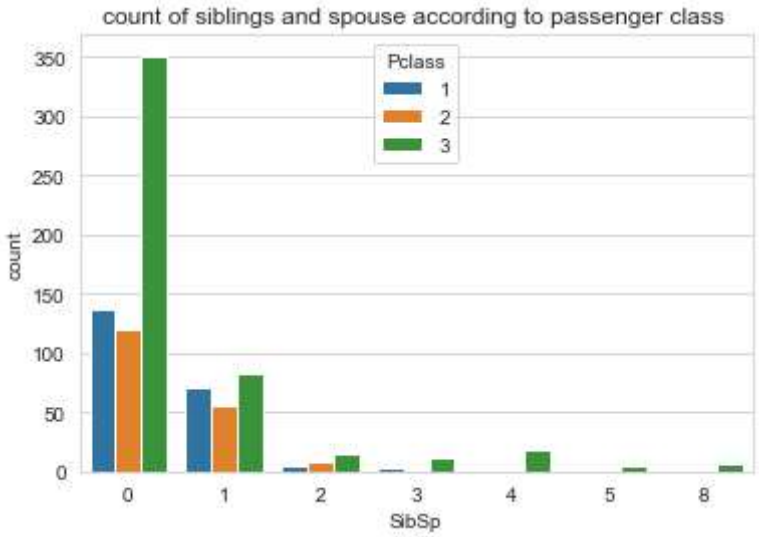
In [10]:
```
plt.title("count of siblings and spouse")
sns.set_style("whitegrid")
sns.countplot(x='SibSp',data=train)
```

Out[10]:
```
<AxesSubplot:title={'center':'count of siblings and spouse'}, xlabel='SibSp', ylabel
='count'>
```



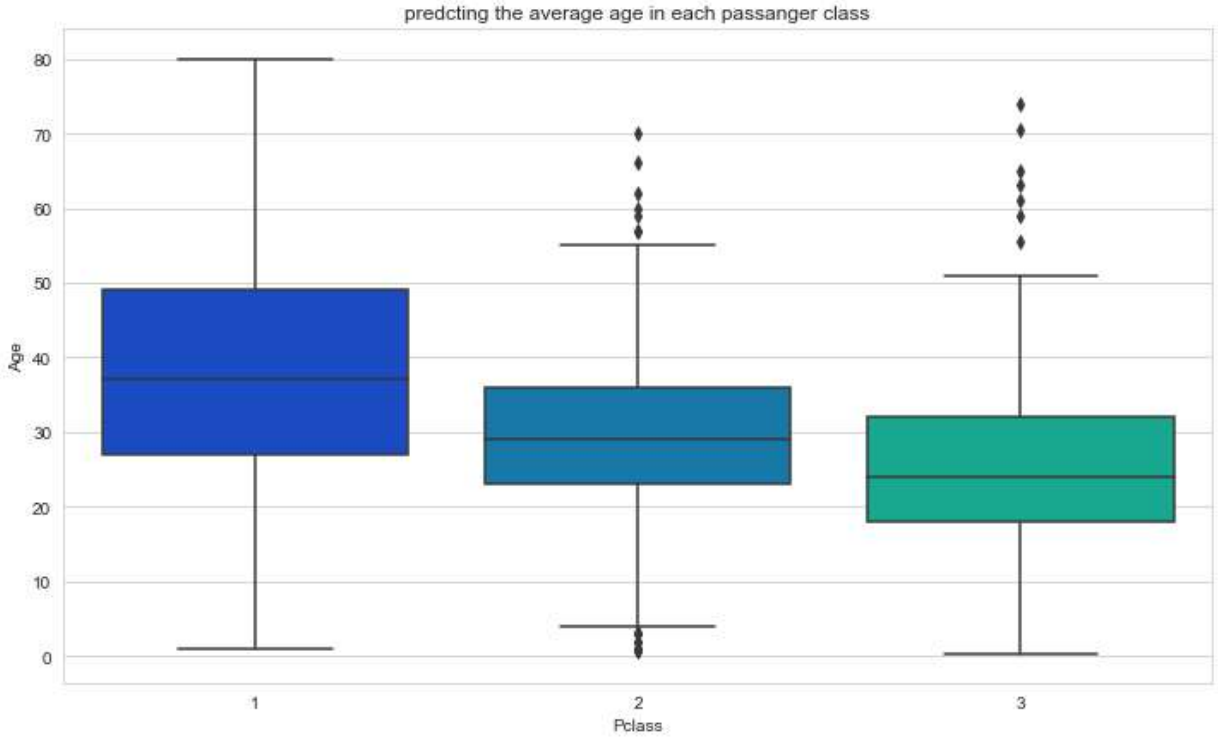
In [11]:
```
plt.title("count of siblings and spouse according to passenger class ")
sns.set_style("whitegrid")
sns.countplot(x='SibSp',hue='Pclass',data=train)
```

Out[11]:
```
<AxesSubplot:title={'center':'count of siblings and spouse according to passenger cl
ass '}, xlabel='SibSp', ylabel='count'>
```



In [12]:
```
plt.figure(figsize=(12,7))
plt.title("predcting the average age in each passanger class")
sns.boxplot(x='Pclass',y='Age',data=train,palette='winter')
```

Out[12]:
```
<AxesSubplot:title={'center':'predcting the average age in each passanger class'}, xlabel='Pclass', ylabel='Age'>
```



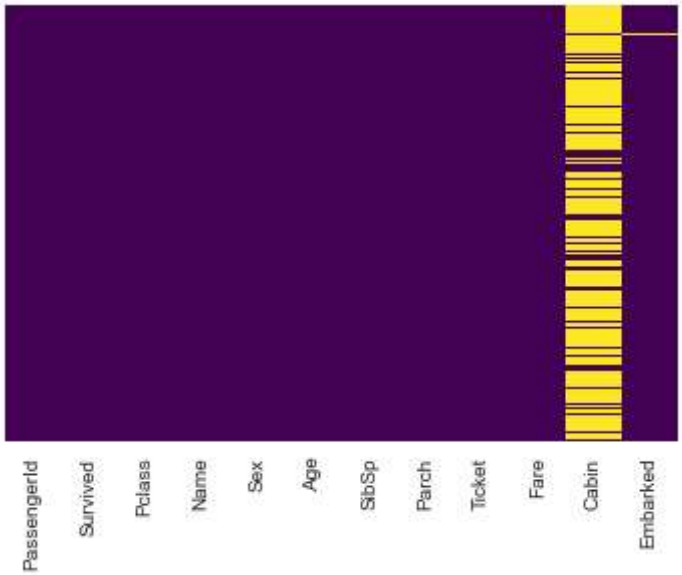
Removing the null values from dataset

In [13]:
```
def impute_age(cols):
    Age=cols[0]
    Pclass=cols[1]
    if pd.isnull(Age):
        if Pclass==1:
            return 37
        elif Pclass ==2:
            return 29
        elif Pclass==3:
            return 24
    else:
        return Age
```

In [14]:
```
train['Age']=train[['Age','Pclass']].apply(impute_age,axis=1)
```

In [15]:
```
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[15]: <AxesSubplot:>



so as by the heatmap the null values from the age data has removed

In [16]:
```
train.drop('Cabin',axis=1,inplace=True)
train
```

Out[16]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Emb
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	

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Emba
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	24.0	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 × 11 columns

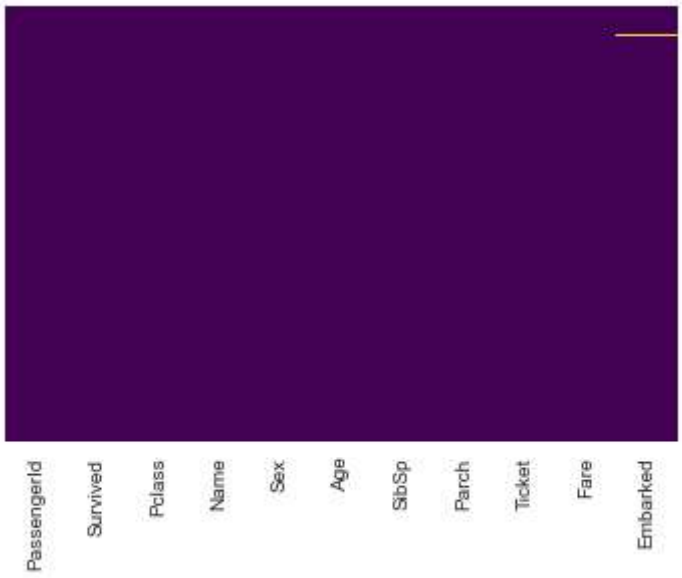
In [17]:
```
train.head()
```

Out[17]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embark
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	

In [18]:
```
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[18]:
```
<AxesSubplot:>
```



converting categorial features

In [19]:
```
train.describe()
```

Out[19]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.066409	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	13.244532	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	22.000000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	26.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	37.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [20]:
```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 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          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  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(4)
memory usage: 76.7+ KB
```

In [21]:
```
pd.get_dummies(train['Embarked'],drop_first=True).head()
```

Out[21]:

	Q	S
0	0	1
1	0	0
2	0	1
3	0	1
4	0	1

In [22]:
```
sex=pd.get_dummies(train['Sex'],drop_first=True)
embark=pd.get_dummies(train['Embarked'],drop_first=True)
```

In [23]:
```
train.drop(['Sex','Embarked','Name','Ticket'],axis=1,inplace=True)
```

In [24]:
```
train.head()
```

Out[24]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
0	1	0	3	22.0	1	0	7.2500
1	2	1	1	38.0	1	0	71.2833
2	3	1	3	26.0	0	0	7.9250
3	4	1	1	35.0	1	0	53.1000
4	5	0	3	35.0	0	0	8.0500

train=pd.concat([train,sex,embark],axis=1)

In [25]:
```
train.head()
```

Out[25]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
0	1	0	3	22.0	1	0	7.2500
1	2	1	1	38.0	1	0	71.2833
2	3	1	3	26.0	0	0	7.9250
3	4	1	1	35.0	1	0	53.1000
4	5	0	3	35.0	0	0	8.0500

Building a Logistic Regression model

Train Test Split

In [26]:
```
train.drop('Survived',axis=1).head()
```

Out[26]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare
0	1	3	22.0	1	0	7.2500
1	2	1	38.0	1	0	71.2833
2	3	3	26.0	0	0	7.9250
3	4	1	35.0	1	0	53.1000

	PassengerId	Pclass	Age	SibSp	Parch	Fare
4	5	3	35.0	0	0	8.0500

In [27]:
```
train['Survived'].head()
```

Out[27]:
```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

In [29]:
```
from sklearn.model_selection import train_test_split
```

In [35]:
```
X_train,X_test,Y_train,Y_test=train_test_split(train.drop('Survived',axis=1),
                                               train['Survived'],test_size=0.30,
                                               random_state=101)
```

Training and predicting

In [36]:
```
from sklearn.linear_model import LogisticRegression
```

In [37]:
```
logmodel=LogisticRegression()
logmodel.fit(X_train,Y_train)
```

Out[37]:
```
LogisticRegression()
```

In [38]:
```
predictions=logmodel.predict(X_test)
```

In [39]:
```
from sklearn.metrics import confusion_matrix
```

In [40]:
```
accuracy=confusion_matrix(Y_test,predictions)
```

In [41]:
```
accuracy
```

Out[41]:
```
array([[133,  21],
       [ 60,  54]], dtype=int64)
```

In [42]:
```
from sklearn.metrics import accuracy_score
```

In [43]:
```
accuracy=accuracy_score(Y_test,predictions)
accuracy
```

Out[43]:
```
0.6977611940298507
```

In [44]:
```
predictions
```

Out[44]:
```
array([1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0,
       0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1,
```

```
0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0,
0, 1, 0, 0], dtype=int64)
```

In [ ]: