Titanic-Machine Learning From Diaster

Titanic Data Set

In [1]:
```
#import the libraries
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:\\titanic\\train.csv')
train.head()
```

Out[2]:

	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 [3]:
```
test=pd.read_csv('D:\\titanic\\test.csv')
test.head()
```

Out[3]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

In [4]:
```
train.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
```

Exploratory Data Analysis

In [5]:
```
#check missing value
```

Missing Data

In [6]:
```
train.isnull()
```

Out[6]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	False	True	False

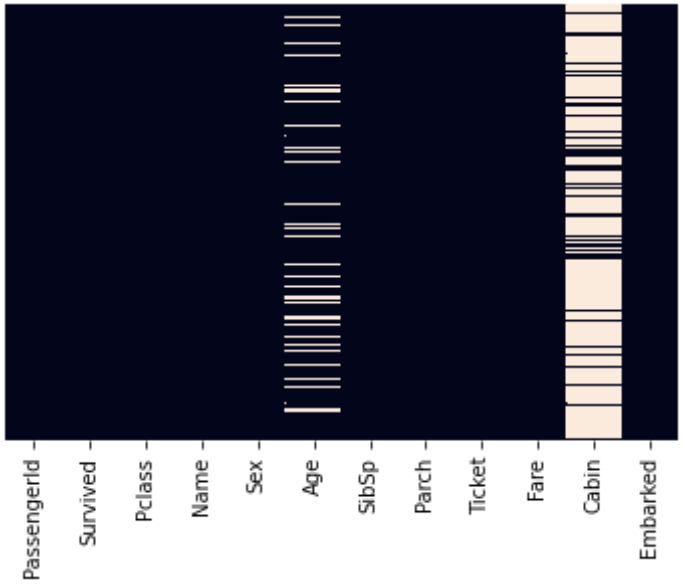
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False	False	False	False	False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

891 rows × 12 columns

In [7]:
```
#using heatmap to check detailed missing values
```

In [8]:
```
sns.heatmap(train.isnull(),cbar=False,yticklabels=False)
```

Out[8]: <AxesSubplot:>

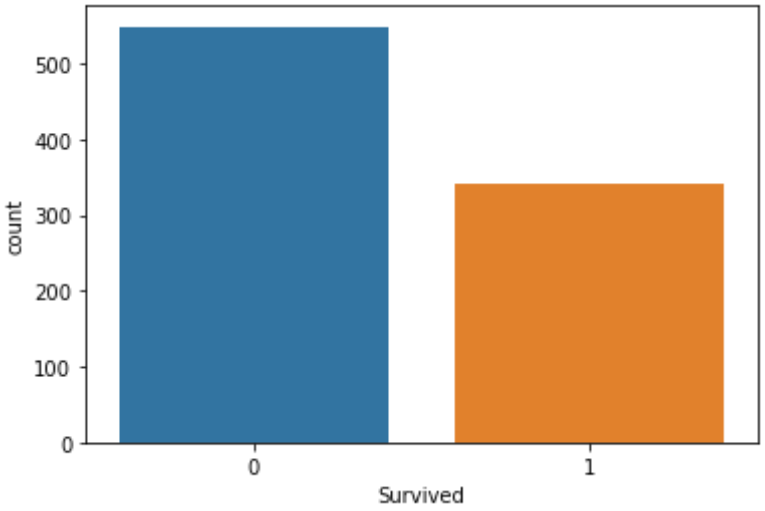


In [ ]:

In [9]:
```
#now counts for how many survived(==1) and not survived(==0)
```

In [10]:
```
sns.countplot(x='Survived',data=train)
```

Out[10]:
```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```

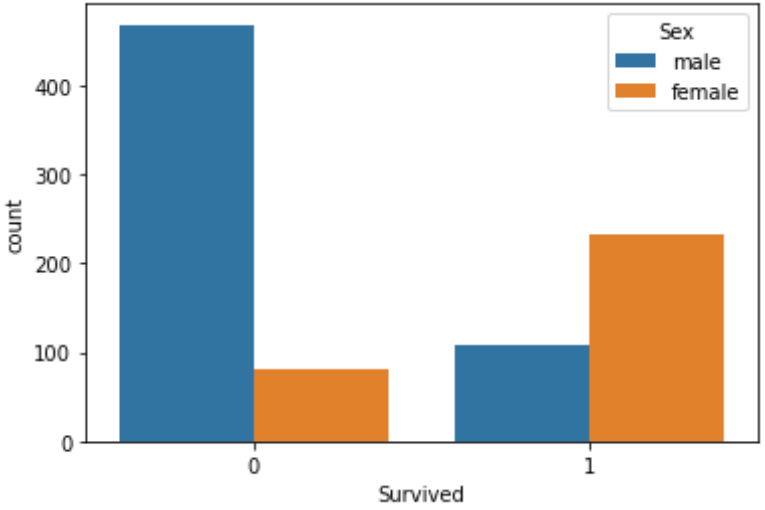


from above plot,we get to know that no. of people were survived less

Now I want to check that no. of male and female survived

In [11]:
```
sns.countplot(x='Survived',hue='Sex',data=train)
```

Out[11]:
```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```

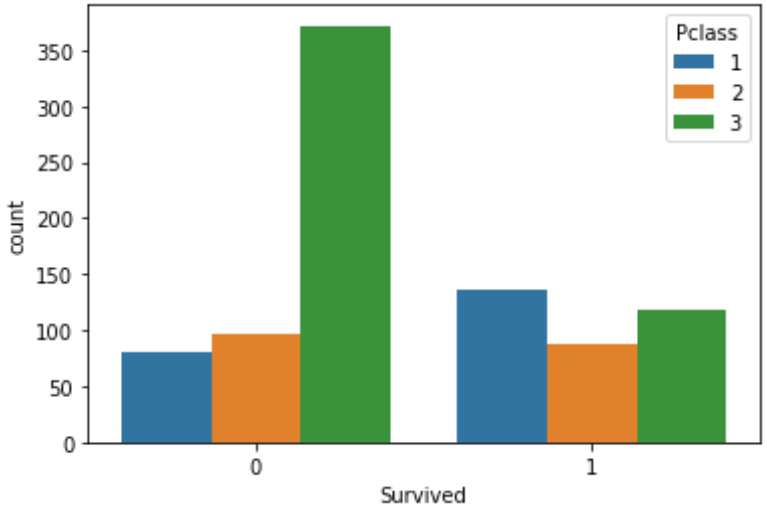


From above plot we get to know that no. of females were survived more than male.

Now i want to know that no. of passangers survived based on there class

In [12]:
```
sns.countplot(x='Survived',hue='Pclass',data=train)
```

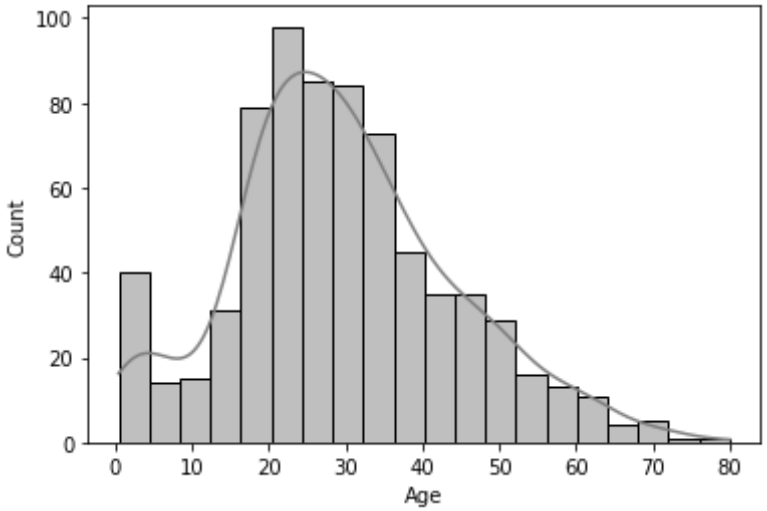
Out[12]:
```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```



check whether Age column follows normal ditribution or not

In [13]:
```
sns.histplot(x='Age',data=train,color='grey',kde=True)
```

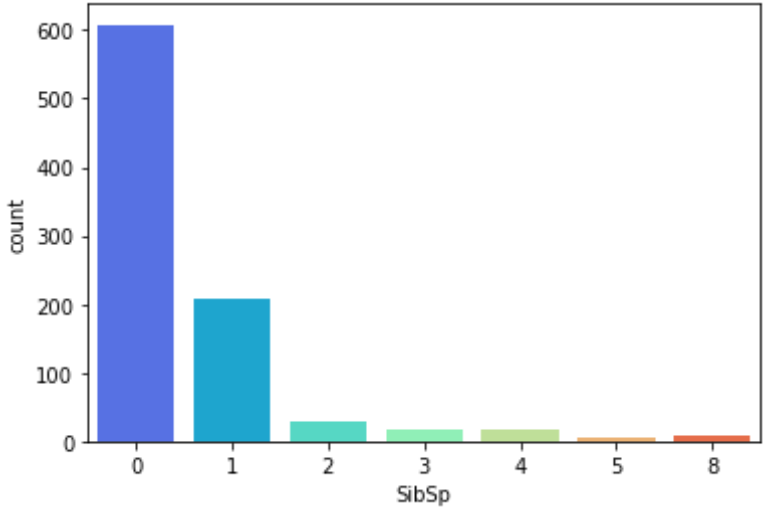
Out[13]:
```
<AxesSubplot:xlabel='Age', ylabel='Count'>
```



count whether sibling/spouse who had survived or not

In [14]:
```
sns.countplot(x='SibSp',data=train,palette='rainbow')
```

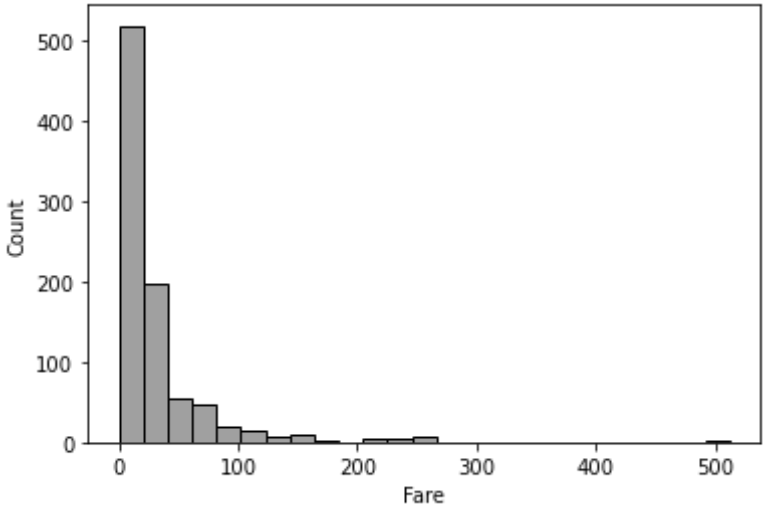
Out[14]:
```
<AxesSubplot:xlabel='SibSp', ylabel='count'>
```



plot a histogram to check who bought tickets

In [15]:
```
sns.histplot(x='Fare',data=train,color='grey',bins=25)
```

Out[15]:
```
<AxesSubplot:xlabel='Fare', ylabel='Count'>
```

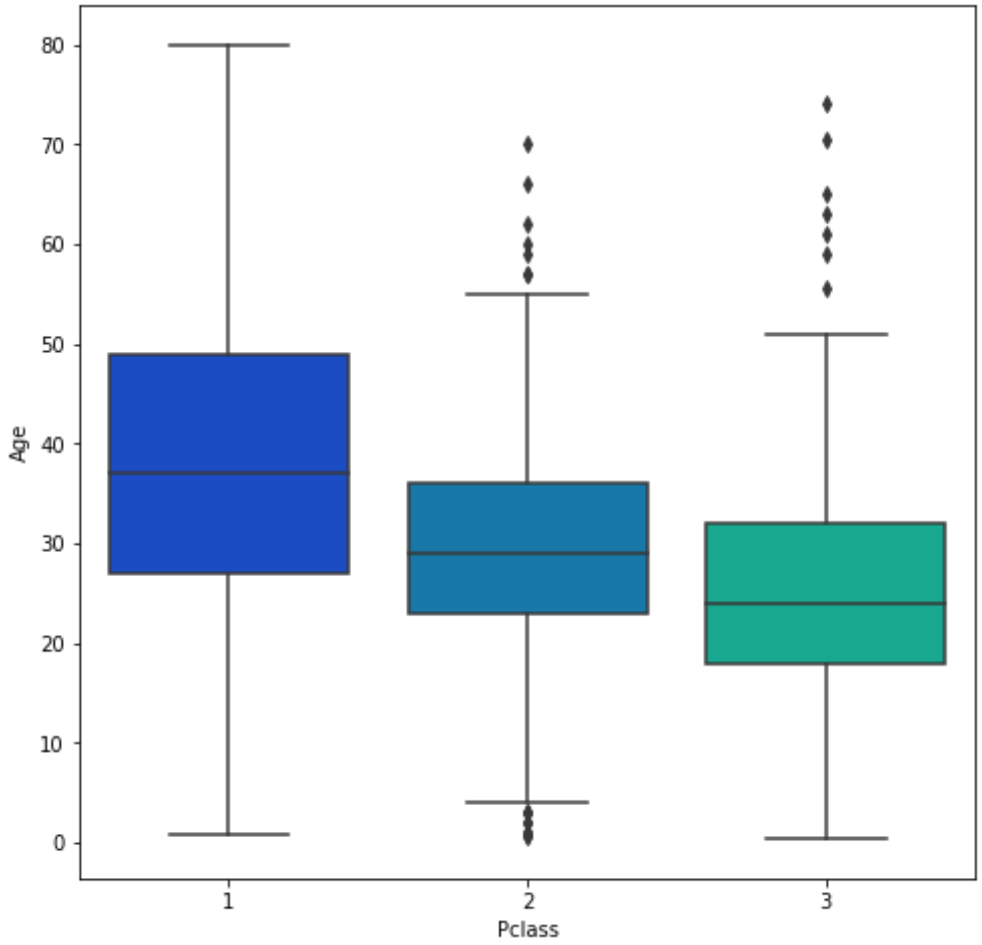


DATA CLEANING

Column "Age" and "Cabin" having null values, instead of droping them, i want to fill with avg value of passanger class of particular age

In [16]:
```
plt.figure(figsize=(8, 8))
sns.boxplot(x='Pclass',y='Age',data=train,palette='winter')
```

Out[16]:
```
<AxesSubplot:xlabel='Pclass', ylabel='Age'>
```



we can clearly see the passanger class 1 is making sense more

In [17]:
```
#fill the null values
```

In [18]:
```
def impute_age(cols):
    Age=cols[0]
    Pclass=cols[1]

    if pd.isnull(Age):
        if Pclass==1:
```

```
            return 38

        elif Pclass==2:
            return 29

        elif Pclass==3:
            return 24

        else:
            return Age
```

In [19]:
```
train['Age']=train[['Age','Pclass']].apply(impute_age,axis=1)
train
```

Out[19]:

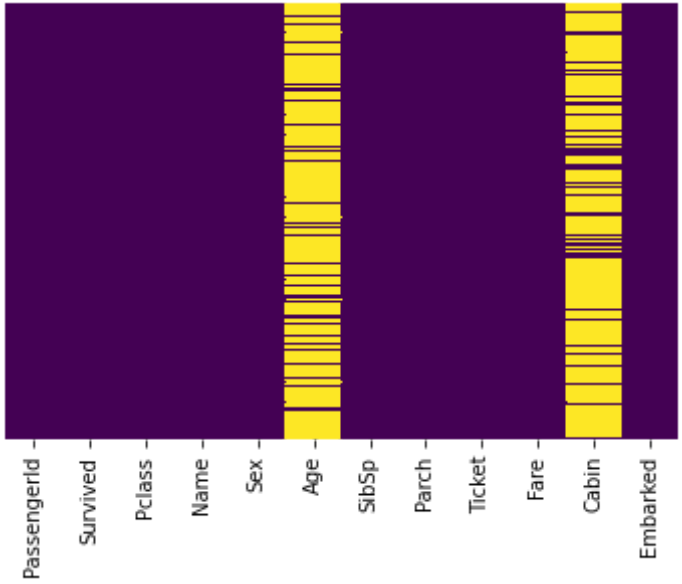
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	NaN	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	NaN	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	NaN	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	NaN	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	NaN	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	NaN	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	NaN	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	24.0	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	NaN	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	NaN	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

In [20]:
```
#lets check heatmap again
```

In [21]:
```
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[21]: <AxesSubplot:>



In [22]:
```
#and drop the cabin column because it has null value high
```

In [23]:
```
train.drop('Cabin',axis=1,inplace=True)
train.head()
```

Out[23]:

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

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
4	5	0	3	Allen, Mr. William Henry	male	NaN	0	0	373450	8.0500	S

In [24]:
```
train.dropna(inplace=True)
```

Converting Categorical into Numerical

In [25]:
```
#column sex and column Embarked has object,so coverting into numerical by one hot coding method
```

In [26]:
```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 177 entries, 5 to 888
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  177 non-null    int64  
 1   Survived     177 non-null    int64  
 2   Pclass       177 non-null    int64  
 3   Name         177 non-null    object 
 4   Sex          177 non-null    object 
 5   Age          177 non-null    float64
 6   SibSp        177 non-null    int64  
 7   Parch        177 non-null    int64  
 8   Ticket       177 non-null    object 
 9   Fare         177 non-null    float64
 10  Embarked     177 non-null    object 
dtypes: float64(2), int64(5), object(4)
memory usage: 16.6+ KB
```

In [27]:
```
Sex=pd.get_dummies(train['Sex'],drop_first=True)
Embarked=pd.get_dummies(train['Embarked'],drop_first=True)
```

In [28]:
```
#also not needed Sex,Embarked,name & ticket column so drop them
```

In [29]:
```
train.drop(['Sex','Embarked','Name','Ticket'],axis=1,inplace=True)
train.head()
```

Out[29]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
5	6	0	3	24.0	0	0	8.4583
17	18	1	2	29.0	0	0	13.0000
19	20	1	3	24.0	0	0	7.2250
26	27	0	3	24.0	0	0	7.2250
28	29	1	3	24.0	0	0	7.8792

Building a logistic regression

Train Test Split

In [30]:
```
#survived is dependent
train.drop('Survived',axis=1).head()
```

Out[30]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare
5	6	3	24.0	0	0	8.4583
17	18	2	29.0	0	0	13.0000
19	20	3	24.0	0	0	7.2250
26	27	3	24.0	0	0	7.2250
28	29	3	24.0	0	0	7.8792

In [ ]:
```
```

In [31]:
```
from sklearn.model_selection import train_test_split
```

In [32]:
```
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)
```

Traing and Prediction

In [33]:
```
from sklearn.linear_model import LogisticRegression
```

In [34]:
```
logmodel=LogisticRegression()
logmodel.fit(X_train,y_train)
```

Out[34]:
```
LogisticRegression()
```

In [35]:
```
predictions=logmodel.predict(X_test)
```

In [36]:
```
!pip3 install -U scikit-learn scipy matplotlib
```

```
Collecting scikit-learn
ERROR: Could not install packages due to an EnvironmentError: [WinError 5] Access is denied: 'C:\\Users\\DELL\\anacon
da3\\Lib\\site-packages\\~cipy\\cluster\\_hierarchy.cp38-win_amd64.pyd'
Consider using the `--user` option or check the permissions.
  Using cached scikit_learn-0.24.2-cp38-cp38-win_amd64.whl (6.9 MB)
Collecting scipy
  Using cached scipy-1.6.3-cp38-cp38-win_amd64.whl (32.7 MB)
Collecting matplotlib
  Using cached matplotlib-3.4.2-cp38-cp38-win_amd64.whl (7.1 MB)
Requirement already satisfied, skipping upgrade: threadpoolctl>=2.0.0 in c:\users\dell\anaconda3\lib\site-packages (f
rom scikit-learn) (2.1.0)
Requirement already satisfied, skipping upgrade: joblib>=0.11 in c:\users\dell\anaconda3\lib\site-packages (from scik
it-learn) (0.17.0)
Requirement already satisfied, skipping upgrade: numpy>=1.13.3 in c:\users\dell\anaconda3\lib\site-packages (from sci
kit-learn) (1.19.2)
Requirement already satisfied, skipping upgrade: cycler>=0.10 in c:\users\dell\anaconda3\lib\site-packages (from matp
lotlib) (0.10.0)
Requirement already satisfied, skipping upgrade: pillow>=6.2.0 in c:\users\dell\anaconda3\lib\site-packages (from mat
plotlib) (8.0.1)
Requirement already satisfied, skipping upgrade: kiwisolver>=1.0.1 in c:\users\dell\anaconda3\lib\site-packages (from
matplotlib) (1.3.0)
Requirement already satisfied, skipping upgrade: pyparsing>=2.2.1 in c:\users\dell\anaconda3\lib\site-packages (from
matplotlib) (2.4.7)
Requirement already satisfied, skipping upgrade: python-dateutil>=2.7 in c:\users\dell\anaconda3\lib\site-packages (f
rom matplotlib) (2.8.1)
Requirement already satisfied, skipping upgrade: six in c:\users\dell\anaconda3\lib\site-packages (from cycler>=0.10-
>matplotlib) (1.15.0)
Installing collected packages: scipy, scikit-learn, matplotlib
  Attempting uninstall: scipy
```

```
    Found existing installation: scipy 1.5.2
    Uninstalling scipy-1.5.2:
      Successfully uninstalled scipy-1.5.2
```

In [40]:
```
from sklearn.metrics import confusion_matrix
```

In [41]:
```
accuracy=confusion_matrix(y_test,predictions)
accuracy
```

Out[41]:
```
array([[35,  6],
       [10,  3]], dtype=int64)
```

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

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

Out[43]:
```
0.7037037037037037
```

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

Out[44]:
```
array([1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 1], dtype=int64)
```

In [ ]:
```

```