In [ ]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Reading or Extracting the data into jupyter Notebook

In [277]:

```
df=pd.read_csv("C:/Users/DELL/Desktop/titanic.data.csv")
```

In [278]:

```
df.head()
```

Out[278]:

	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

Exploratory Data Analysis (EDA)

Finding the Null Values

In [279]:
```
df.isnull()
```

Out[279]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emb
0	False	False	False	False	False	False	False	False	False	False	True	
1	False	False	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	True	
3	False	False	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	True	
5	False	False	False	False	False	True	False	False	False	False	True	
6	False	False	False	False	False	False	False	False	False	False	False	
7	False	False	False	False	False	False	False	False	False	False	True	
8	False	False	False	False	False	False	False	False	False	False	True	
9	False	False	False	False	False	False	False	False	False	False	True	
10	False	False	False	False	False	False	False	False	False	False	False	
11	False	False	False	False	False	False	False	False	False	False	False	
12	False	False	False	False	False	False	False	False	False	False	True	
13	False	False	False	False	False	False	False	False	False	False	True	
14	False	False	False	False	False	False	False	False	False	False	True	
15	False	False	False	False	False	False	False	False	False	False	True	
16	False	False	False	False	False	False	False	False	False	False	True	
17	False	False	False	False	False	True	False	False	False	False	True	
18	False	False	False	False	False	False	False	False	False	False	True	
19	False	False	False	False	False	True	False	False	False	False	True	
20	False	False	False	False	False	False	False	False	False	False	True	
21	False	False	False	False	False	False	False	False	False	False	False	
22	False	False	False	False	False	False	False	False	False	False	True	
23	False	False	False	False	False	False	False	False	False	False	False	
24	False	False	False	False	False	False	False	False	False	False	True	
25	False	False	False	False	False	False	False	False	False	False	True	
26	False	False	False	False	False	True	False	False	False	False	True	
27	False	False	False	False	False	False	False	False	False	False	False	
28	False	False	False	False	False	True	False	False	False	False	True	
29	False	False	False	False	False	True	False	False	False	False	True	
...	...	...	...	...	...	...	...	...	...	...	...	
861	False	False	False	False	False	False	False	False	False	False	True	
862	False	False	False	False	False	False	False	False	False	False	False	
863	False	False	False	False	False	True	False	False	False	False	True	

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emb
864	False	False	False	False	False	False	False	False	False	False	True	
865	False	False	False	False	False	False	False	False	False	False	True	
866	False	False	False	False	False	False	False	False	False	False	True	
867	False	False	False	False	False	False	False	False	False	False	False	
868	False	False	False	False	False	True	False	False	False	False	True	
869	False	False	False	False	False	False	False	False	False	False	True	
870	False	False	False	False	False	False	False	False	False	False	True	
871	False	False	False	False	False	False	False	False	False	False	False	
872	False	False	False	False	False	False	False	False	False	False	False	
873	False	False	False	False	False	False	False	False	False	False	True	
874	False	False	False	False	False	False	False	False	False	False	True	
875	False	False	False	False	False	False	False	False	False	False	True	
876	False	False	False	False	False	False	False	False	False	False	True	
877	False	False	False	False	False	False	False	False	False	False	True	
878	False	False	False	False	False	True	False	False	False	False	True	
879	False	False	False	False	False	False	False	False	False	False	False	
880	False	False	False	False	False	False	False	False	False	False	True	
881	False	False	False	False	False	False	False	False	False	False	True	
882	False	False	False	False	False	False	False	False	False	False	True	
883	False	False	False	False	False	False	False	False	False	False	True	
884	False	False	False	False	False	False	False	False	False	False	True	
885	False	False	False	False	False	False	False	False	False	False	True	
886	False	False	False	False	False	False	False	False	False	False	True	
887	False	False	False	False	False	False	False	False	False	False	False	
888	False	False	False	False	False	True	False	False	False	False	True	
889	False	False	False	False	False	False	False	False	False	False	False	
890	False	False	False	False	False	False	False	False	False	False	True	

891 rows × 12 columns

In [280]:

```
df.isnull().sum()
```

Out[280]:

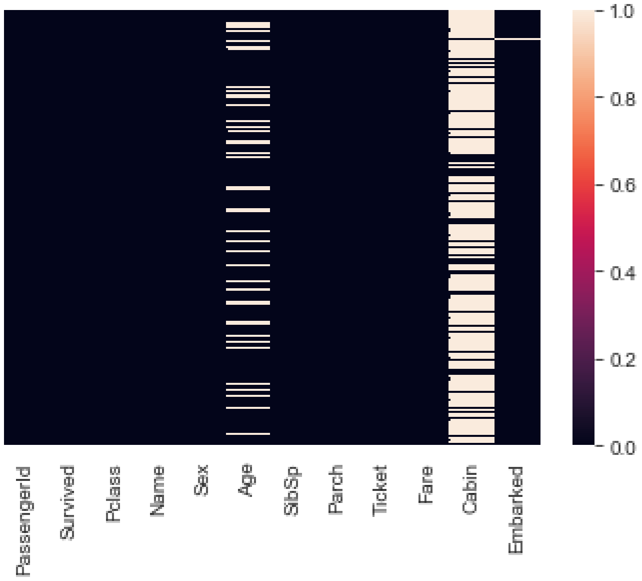
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [281]:

```
sns.heatmap(df.isnull(),yticklabels=False)
```

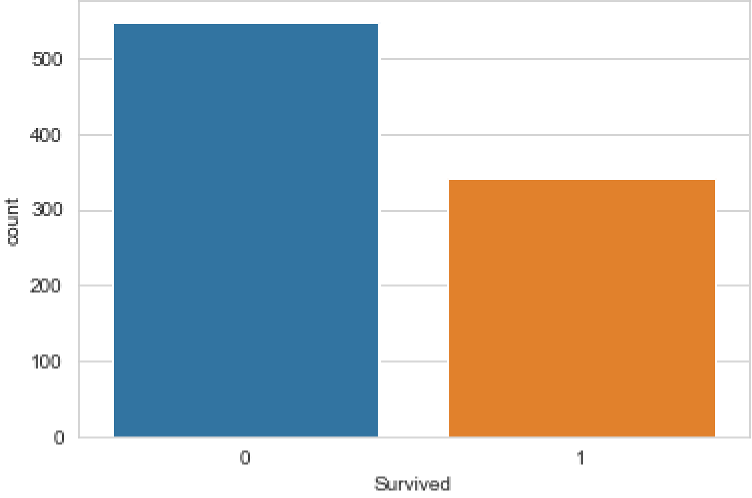
Out[281]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1b6fa7a9978>
```



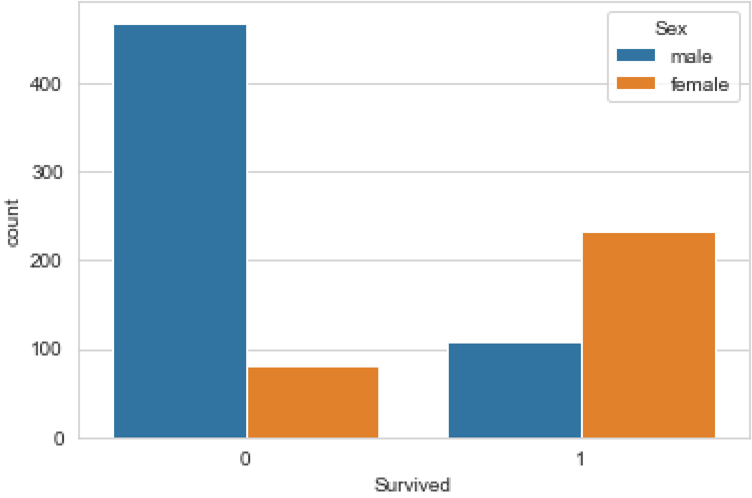
In [282]:
```
sns.set_style("whitegrid")
sns.countplot(x="Survived",data=df)
```

Out[282]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1b6fa77c160>
```



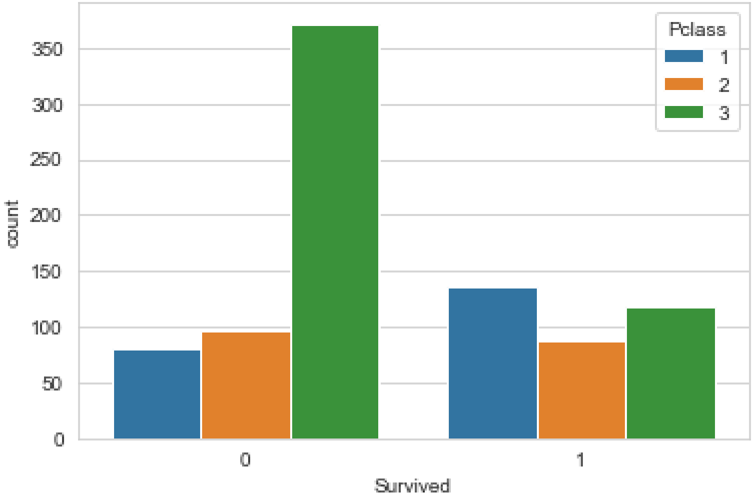
In [283]:
```
sns.set_style("whitegrid")
sns.countplot(x="Survived",hue="Sex",data=df)
```

Out[283]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1b6fb88c080>
```



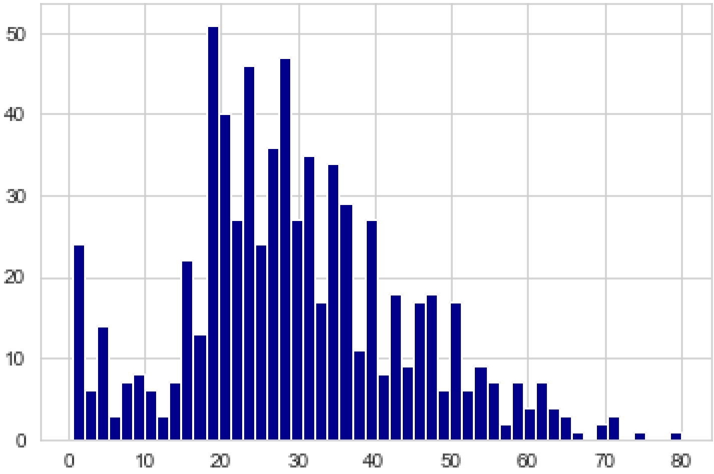
In [284]:
```
sns.set_style("whitegrid")
sns.countplot("Survived",hue="Pclass",data=df)
```

Out[284]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1b6fb8db080>
```



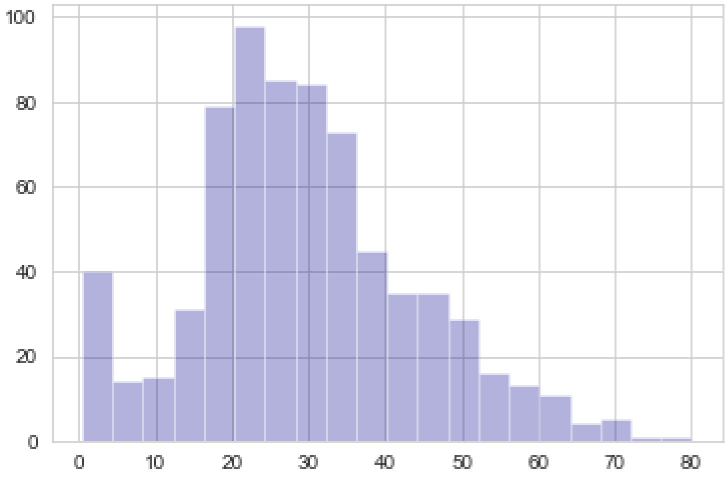
In [285]:
```
df["Age"].hist(bins=50,color="darkblue")
```

Out[285]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1b6fb9300b8>
```



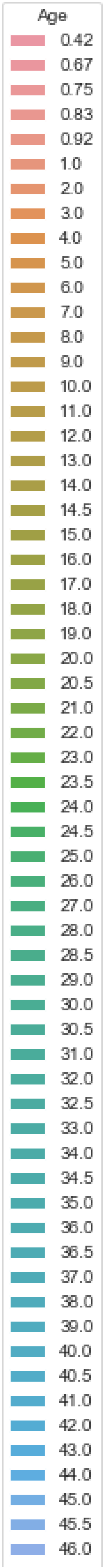
In [286]:
```
df["Age"].hist(bins=20,color="darkblue",alpha=0.3)
```

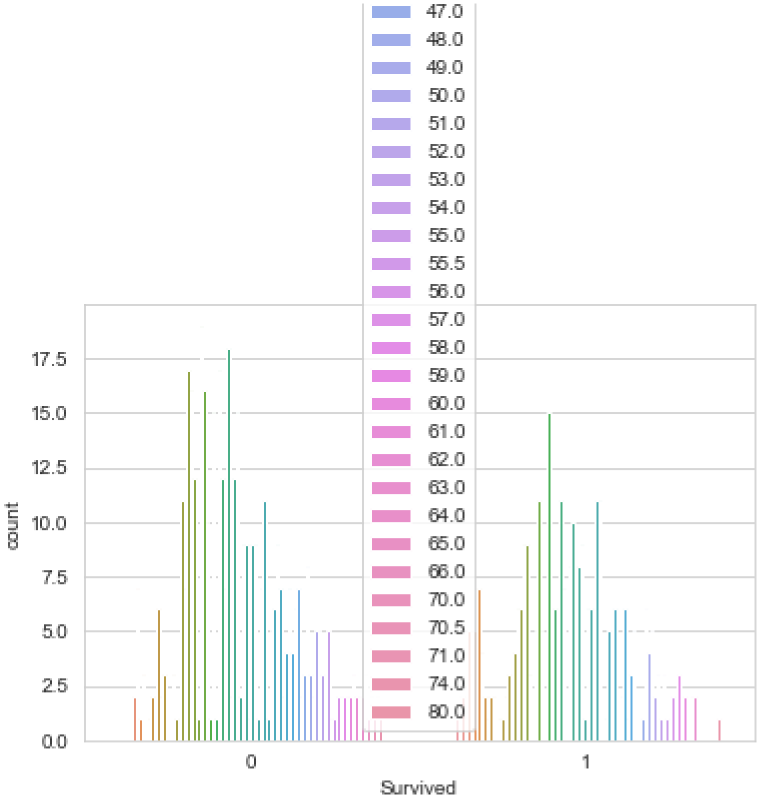
Out[286]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1b6fb9fea90>
```



In [287]:
```
sns.set_style("whitegrid")
sns.countplot(x="Survived",hue="Age",data=df)
```

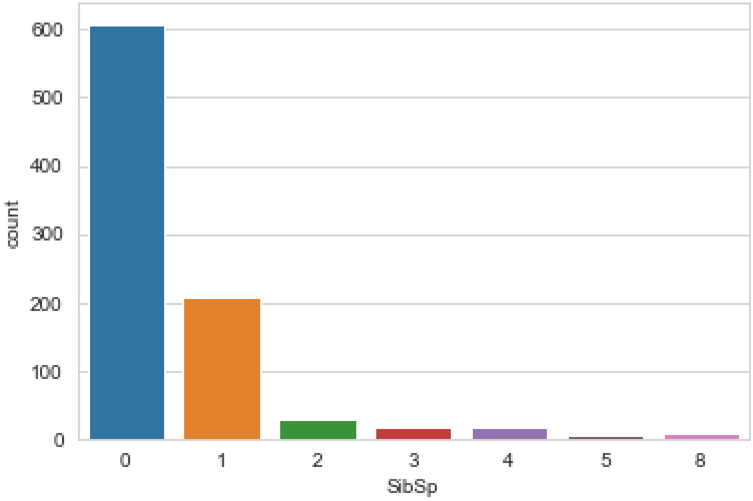
Out[287]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1b6fba482b0>
```





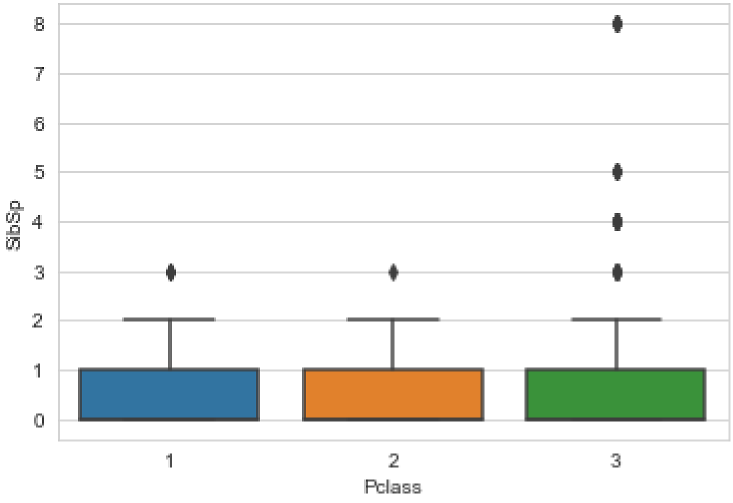
```
In [288]: sns.countplot(x="SibSp",data=df)
```

```
Out[288]: <matplotlib.axes._subplots.AxesSubplot at 0x1b6fbd6c400>
```



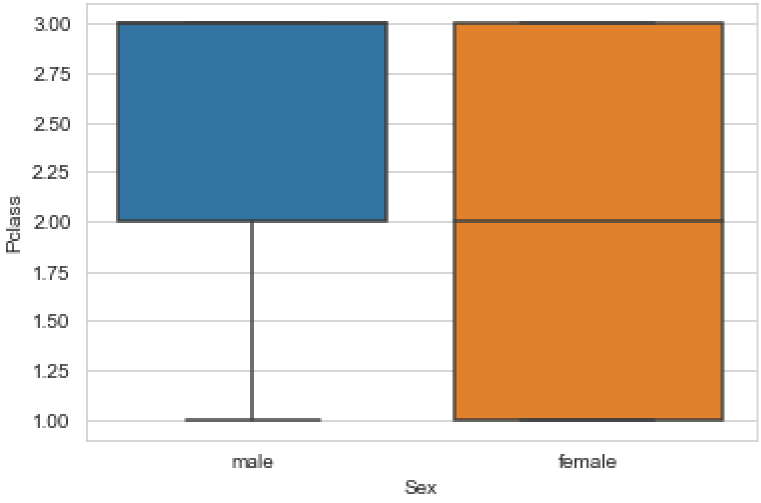
```
In [289]: sns.boxplot(x="Pclass",y="SibSp",data=df)
```

```
Out[289]: <matplotlib.axes._subplots.AxesSubplot at 0x1b6fbd97fd0>
```



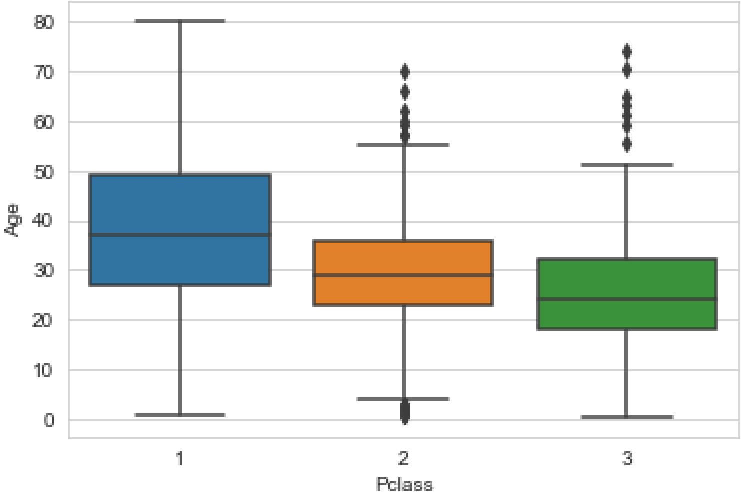
```
In [290]: sns.boxplot(x="Sex",y="Pclass",data=df)
```

```
Out[290]: <matplotlib.axes._subplots.AxesSubplot at 0x1b6fc0756a0>
```



In [291]:
```
sns.boxplot(x="Pclass",y='Age',data=df)
```

Out[291]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1b6fc0d7320>
```



Seperating Numerical and Categorical data or values into some variables

In [292]:
```
df_cat=df.select_dtypes(include=[object])
df_num=df.select_dtypes(include=[np.number])
```

In [293]:
```
df_cat.head()
```

Out[293]:

	Name	Sex	Ticket	Cabin	Embarked
0	Braund, Mr. Owen Harris	male	A/5 21171	NaN	S
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	PC 17599	C85	C
2	Heikkinen, Miss. Laina	female	STON/O2. 3101282	NaN	S
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	113803	C123	S
4	Allen, Mr. William Henry	male	373450	NaN	S

In [294]:

```
df_num.head()
```

Out[294]:

	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

In [295]:

```
df_cat.isnull()
```

Out[295]:

	Name	Sex	Ticket	Cabin	Embarked
0	False	False	False	True	False
1	False	False	False	False	False
2	False	False	False	True	False
3	False	False	False	False	False
4	False	False	False	True	False
5	False	False	False	True	False
6	False	False	False	False	False
7	False	False	False	True	False
8	False	False	False	True	False
9	False	False	False	True	False
10	False	False	False	False	False
11	False	False	False	False	False
12	False	False	False	True	False
13	False	False	False	True	False
14	False	False	False	True	False
15	False	False	False	True	False
16	False	False	False	True	False
17	False	False	False	True	False
18	False	False	False	True	False
19	False	False	False	True	False
20	False	False	False	True	False
21	False	False	False	False	False
22	False	False	False	True	False
23	False	False	False	False	False
24	False	False	False	True	False
25	False	False	False	True	False
26	False	False	False	True	False
27	False	False	False	False	False
28	False	False	False	True	False
29	False	False	False	True	False
...	...	...	...	...	...
861	False	False	False	True	False
862	False	False	False	False	False
863	False	False	False	True	False

	Name	Sex	Ticket	Cabin	Embarked
864	False	False	False	True	False
865	False	False	False	True	False
866	False	False	False	True	False
867	False	False	False	False	False
868	False	False	False	True	False
869	False	False	False	True	False
870	False	False	False	True	False
871	False	False	False	False	False
872	False	False	False	False	False
873	False	False	False	True	False
874	False	False	False	True	False
875	False	False	False	True	False
876	False	False	False	True	False
877	False	False	False	True	False
878	False	False	False	True	False
879	False	False	False	False	False
880	False	False	False	True	False
881	False	False	False	True	False
882	False	False	False	True	False
883	False	False	False	True	False
884	False	False	False	True	False
885	False	False	False	True	False
886	False	False	False	True	False
887	False	False	False	False	False
888	False	False	False	True	False
889	False	False	False	False	False
890	False	False	False	True	False

891 rows × 5 columns

In [296]:

```
df_num.isnull()
```

Out[296]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
5	False	False	False	True	False	False	False
6	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False
10	False	False	False	False	False	False	False
11	False	False	False	False	False	False	False
12	False	False	False	False	False	False	False
13	False	False	False	False	False	False	False
14	False	False	False	False	False	False	False
15	False	False	False	False	False	False	False
16	False	False	False	False	False	False	False
17	False	False	False	True	False	False	False
18	False	False	False	False	False	False	False
19	False	False	False	True	False	False	False
20	False	False	False	False	False	False	False
21	False	False	False	False	False	False	False
22	False	False	False	False	False	False	False
23	False	False	False	False	False	False	False
24	False	False	False	False	False	False	False
25	False	False	False	False	False	False	False
26	False	False	False	True	False	False	False
27	False	False	False	False	False	False	False
28	False	False	False	True	False	False	False
29	False	False	False	True	False	False	False
...	...	...	...	...	...	...	...
861	False	False	False	False	False	False	False
862	False	False	False	False	False	False	False
863	False	False	False	True	False	False	False

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
864	False	False	False	False	False	False	False
865	False	False	False	False	False	False	False
866	False	False	False	False	False	False	False
867	False	False	False	False	False	False	False
868	False	False	False	True	False	False	False
869	False	False	False	False	False	False	False
870	False	False	False	False	False	False	False
871	False	False	False	False	False	False	False
872	False	False	False	False	False	False	False
873	False	False	False	False	False	False	False
874	False	False	False	False	False	False	False
875	False	False	False	False	False	False	False
876	False	False	False	False	False	False	False
877	False	False	False	False	False	False	False
878	False	False	False	True	False	False	False
879	False	False	False	False	False	False	False
880	False	False	False	False	False	False	False
881	False	False	False	False	False	False	False
882	False	False	False	False	False	False	False
883	False	False	False	False	False	False	False
884	False	False	False	False	False	False	False
885	False	False	False	False	False	False	False
886	False	False	False	False	False	False	False
887	False	False	False	False	False	False	False
888	False	False	False	True	False	False	False
889	False	False	False	False	False	False	False
890	False	False	False	False	False	False	False

891 rows × 7 columns

In [297]:

```
df_cat.isnull().sum()
```

Out[297]:

```
Name          0
Sex           0
Ticket        0
Cabin       687
Embarked      2
dtype: int64
```

In [298]:

```
df_num.isnull().sum()
```

Out[298]:

```
PassengerId      0
Survived         0
Pclass           0
Age            177
SibSp            0
Parch            0
Fare             0
dtype: int64
```

Removing the null values by using Fillna() Function

In [299]:

```
df_cat.Cabin.fillna(df_cat.Cabin.value_counts().idxmax(),inplace=True)
df_cat.Embarked.fillna(df_cat.Embarked.value_counts().idxmax(),inplace=True)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py:6130: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stab
le/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pandas-doc
s/stable/indexing.html#indexing-view-versus-copy)
  self._update_inplace(new_data)
```

In [300]:

```
df_cat.isnull().sum()
```

Out[300]:

```
Name        0
Sex         0
Ticket      0
Cabin       0
Embarked    0
dtype: int64
```

In [301]:

```
df_num.Age.fillna(df_num.Age.value_counts().mean(),inplace=True)
```

In [302]:

```
df_num.isnull().sum()
```

Out[302]:

```
PassengerId    0
Survived       0
Pclass         0
Age            0
SibSp          0
Parch          0
Fare           0
dtype: int64
```

Concatinating both Categorical and Numerical Values

In [303]:

```
df=pd.concat([df_cat,df_num],axis=1)
```

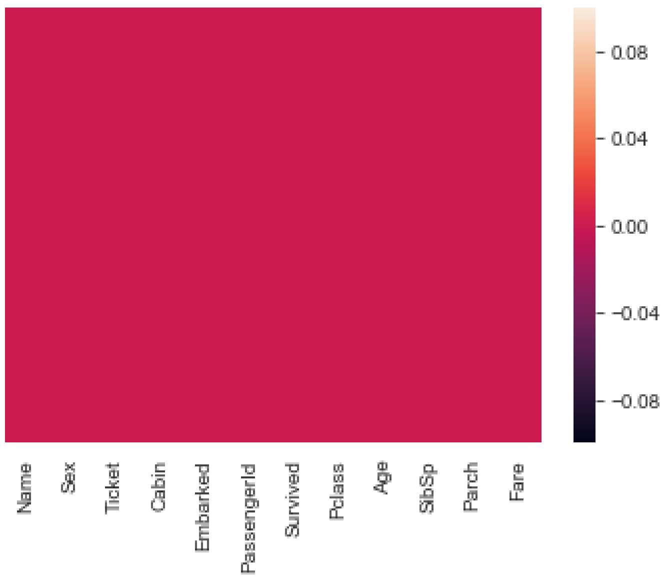
In [304]:
```
df.head()
```

Out[304]:

	Name	Sex	Ticket	Cabin	Embarked	PassengerId	Survived	Pclass	Age	SibSp	Pa
0	Braund, Mr. Owen Harris	male	A/5 21171	C23 C25 C27	S	1	0	3	22.0	1	
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	PC 17599	C85	C	2	1	1	38.0	1	
2	Heikkinen, Miss. Laina	female	STON/O2. 3101282	C23 C25 C27	S	3	1	3	26.0	0	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	113803	C123	S	4	1	1	35.0	1	
4	Allen, Mr. William Henry	male	373450	C23 C25 C27	S	5	0	3	35.0	0	

In [118]:
```
sns.heatmap(df.isnull(),yticklabels=False)
```

Out[118]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1b6f76f2f60>
```



In [119]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
Name           891 non-null object
Sex            891 non-null object
Ticket         891 non-null object
Cabin          891 non-null object
Embarked       891 non-null object
PassengerId    891 non-null int64
Survived       891 non-null int64
Pclass         891 non-null int64
Age            891 non-null float64
SibSp          891 non-null int64
Parch          891 non-null int64
Fare           891 non-null float64
dtypes: float64(2), int64(5), object(5)
memory usage: 83.6+ KB
```

In [120]:

```
df.describe()
```

Out[120]:

	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	25.411093	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	15.598322	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	8.113636	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	24.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	35.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

Converting Categorical data into Numerical data by LabelEncoder for model

In [121]:

```
from sklearn.preprocessing import LabelEncoder
```

In [122]:

```
encoder=LabelEncoder()
```

In [123]:

```
df_cat=df_cat.apply(encoder.fit_transform)
```

In [124]:

```
df=pd.concat([df_cat,df_num],axis=1)
```

our data is set for model creation

Logistic Regression

In [125]:
```
df.head()
```

Out[125]:

	Name	Sex	Ticket	Cabin	Embarked	PassengerId	Survived	Pclass	Age	SibSp	Parch	F
0	108	1	523	63	2	1	0	3	22.0	1	0	7.2
1	190	0	596	81	0	2	1	1	38.0	1	0	71.2
2	353	0	669	63	2	3	1	3	26.0	0	0	7.9
3	272	0	49	55	2	4	1	1	35.0	1	0	53.1
4	15	1	472	63	2	5	0	3	35.0	0	0	8.0

In [143]:
```
df.drop(["Name","Embarked","Sex"],axis=1,inplace=True)
```

In [144]:
```
df.head()
```

Out[144]:

	Ticket	Cabin	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
0	523	63	1	0	3	22.0	1	0	7.2500
1	596	81	2	1	1	38.0	1	0	71.2833
2	669	63	3	1	3	26.0	0	0	7.9250
3	49	55	4	1	1	35.0	1	0	53.1000
4	472	63	5	0	3	35.0	0	0	8.0500

In [146]:
```
x=df.iloc[:,[0,1,2,4,5,7,8]].values
y=df.iloc[:,3].values
```

In [147]:
```
from sklearn.model_selection import train_test_split
```

In [148]:
```
 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.30, rando
```

In [149]:
```
from sklearn.linear_model import LogisticRegression
```

In [150]:
```
lr=LogisticRegression()
```

In [151]:
```
lr.fit(x_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:43
3: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a
solver to silence this warning.
  FutureWarning)
```

Out[151]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='warn',
          tol=0.0001, verbose=0, warm_start=False)
```

In [152]:
```
predictions=lr.predict(x_test)
```

In [153]:
```
predictions
```

Out[153]:
```
array([1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 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, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0,
       0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0,
       1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0,
       0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0], dtype=int64)
```

In [154]:
```
from sklearn.metrics import confusion_matrix
```

In [155]:
```
cm=confusion_matrix(y_test,predictions)
```

In [156]:
```
cm
```

Out[156]:
```
array([[150,  12],
       [ 72,  34]], dtype=int64)
```

In [157]:
```
from sklearn.metrics import classification_report
```

In [158]:
```
cr=classification_report(y_test,predictions)
```

In [159]:
```
cr
```

Out[159]:
```
'              precision    recall  f1-score   support\n\n           0       0.
68      0.93      0.78       162\n           1       0.74      0.32      0.45
106\n\n   micro avg       0.69      0.69      0.69       268\n   macro avg
0.71      0.62      0.61       268\nweighted avg       0.70      0.69      0.65
268\n'
```

In [160]:
```
from sklearn.metrics import accuracy_score
```

In [161]:
```
score=accuracy_score(y_test,predictions)
```

In [162]:
```
score
```

Out[162]:
```
0.6865671641791045
```

Decision Tree

In [163]:
```
df.head()
```

Out[163]:

	Ticket	Cabin	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
0	523	63	1	0	3	22.0	1	0	7.2500
1	596	81	2	1	1	38.0	1	0	71.2833
2	669	63	3	1	3	26.0	0	0	7.9250
3	49	55	4	1	1	35.0	1	0	53.1000
4	472	63	5	0	3	35.0	0	0	8.0500

In [164]:
```
x1=df.iloc[:,[0,1,2,4,5,6,7,8]].values
y=df.iloc[:,3].values
```

In [165]:
```
from sklearn.model_selection import train_test_split
```

In [166]:
```
x1_train, x1_test, y_train, y_test = train_test_split(x1, y, test_size=0.30, ran
```

In [167]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [172]:
```
dt=DecisionTreeClassifier()
```

In [173]:
```
dt.fit(x1_train,y_train)
```

Out[173]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
```

In [174]:
```
predictions1=dt.predict(x1_test)
```

```
In [175]: predictions1
```

```
Out[175]: array([0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1,
                 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0,
                 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
                 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0,
                 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0,
                 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1,
                 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1,
                 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
                 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
                 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0,
                 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0,
                 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0,
                 0, 0, 0, 1], dtype=int64)
```

```
In [176]: from sklearn.metrics import confusion_matrix
```

```
In [177]: cm1=confusion_matrix(y_test,predictions1)
```

```
In [178]: cm1
```

```
Out[178]: array([[116,  46],
                 [ 45,  61]], dtype=int64)
```

```
In [179]: from sklearn.metrics import classification_report
```

```
In [180]: cr1=classification_report(y_test,predictions1)
```

```
In [181]: cr1
```

```
Out[181]: '              precision    recall  f1-score   support\n\n           0       0.
          72      0.72      0.72       162\n           1       0.57      0.58      0.57
          106\n\n   micro avg       0.66      0.66      0.66       268\n   macro avg
          0.65      0.65      0.65       268\nweighted avg       0.66      0.66      0.66
          268\n'
```

```
In [182]: from sklearn.metrics import accuracy_score
```

```
In [183]: score1=accuracy_score(y_test,predictions1)
```

```
In [184]: score1
```

```
Out[184]: 0.6604477611940298
```

Random Forest

In [185]:
```
df.head()
```

Out[185]:

	Ticket	Cabin	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
0	523	63	1	0	3	22.0	1	0	7.2500
1	596	81	2	1	1	38.0	1	0	71.2833
2	669	63	3	1	3	26.0	0	0	7.9250
3	49	55	4	1	1	35.0	1	0	53.1000
4	472	63	5	0	3	35.0	0	0	8.0500

In [187]:
```
x2=df.iloc[:,[0,1,2,4,5,6,7,8]].values
y=df.iloc[:,[3]].values
```

In [188]:
```
from sklearn.model_selection import train_test_split
```

In [189]:
```
x2_train, x2_test, y_train, y_test = train_test_split(x2, y, test_size=0.30, ran
```

In [190]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [191]:
```
rf=RandomForestClassifier()
```

In [192]:
```
rf.fit(x2_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:246: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: DataConvers
ionWarning: A column-vector y was passed when a 1d array was expected. Please c
hange the shape of y to (n_samples,), for example using ravel().
  """Entry point for launching an IPython kernel.
```

Out[192]:
```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=None,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)
```

In [194]:
```
prediction2=rf.predict(x2_test)
```

In [195]:

```
prediction2
```

Out[195]:

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

In [196]:

```
from sklearn.preprocessing import StandardScaler
```

In [197]:

```
sc=StandardScaler()
```

In [198]:

```
x2_train=sc.fit_transform(x2_train)
x2_test=sc.transform(x2_test)
```

In [199]:

```
sc
```

Out[199]:

```
StandardScaler(copy=True, with_mean=True, with_std=True)
```

In [200]:

```
from sklearn.metrics import confusion_matrix
```

In [201]:

```
cm2=confusion_matrix(y_test,prediction2)
```

In [202]:

```
cm2
```

Out[202]:

```
array([[141,  21],
       [ 54,  52]], dtype=int64)
```

In [203]:

```
from sklearn.metrics import classification_report
```

In [204]:

```
cr2=classification_report(y_test,prediction2)
```

In [205]:

```
cr2
```

Out[205]:

```
'              precision    recall  f1-score   support\n\n           0       0.
72      0.87      0.79       162\n           1       0.71      0.49      0.58
106\n\n   micro avg       0.72      0.72      0.72       268\n   macro avg
0.72      0.68      0.69       268\nweighted avg       0.72      0.72      0.71
268\n'
```

In [206]:

```
from sklearn.metrics import accuracy_score
```

In [207]:

```
score2=accuracy_score(y_test,prediction2)
```

In [208]:
```
score2
```

Out[208]:
```
0.7201492537313433
```

K Nearest Neighbour (KNN)

In [209]:
```
df.head()
```

Out[209]:

	Ticket	Cabin	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
0	523	63	1	0	3	22.0	1	0	7.2500
1	596	81	2	1	1	38.0	1	0	71.2833
2	669	63	3	1	3	26.0	0	0	7.9250
3	49	55	4	1	1	35.0	1	0	53.1000
4	472	63	5	0	3	35.0	0	0	8.0500

In [210]:
```
x3=df.iloc[:,[0,1,2,4,5,6,7,8]].values
y=df.iloc[:,3].values
```

In [211]:
```
from sklearn.model_selection import train_test_split
```

In [212]:
```
x3_train, x3_test, y_train, y_test = train_test_split(x3, y, test_size=0.30, ran
```

In [213]:
```
from sklearn.neighbors import KNeighborsClassifier
```

In [214]:
```
knn=KNeighborsClassifier()
```

In [215]:
```
knn.fit(x3_train,y_train)
```

Out[215]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=None, n_neighbors=5, p=2,
           weights='uniform')
```

In [216]:
```
from sklearn.preprocessing import StandardScaler
```

In [217]:
```
sc3=StandardScaler()
```

In [218]:
```
x3_train=sc3.fit_transform(x3_train)
x3_test=sc3.transform(x3_test)
```

In [219]:
```
sc3
```

Out[219]:
```
StandardScaler(copy=True, with_mean=True, with_std=True)
```

In [220]:
```
prediction3=knn.predict(x3_test)
```

In [221]:
```
prediction3
```

Out[221]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1], dtype=int64)
```

In [222]:
```
from sklearn.metrics import confusion_matrix
```

In [223]:
```
cm3=confusion_matrix(y_test,prediction3)
```

In [224]:
```
cm3
```

Out[224]:
```
array([[  0, 162],
       [  0, 106]], dtype=int64)
```

In [225]:
```
from sklearn.metrics import classification_report
```

In [226]:
```
cr3=classification_report(y_test,prediction3)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:11
43: UndefinedMetricWarning: Precision and F-score are ill-defined and being set
to 0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:11
43: UndefinedMetricWarning: Precision and F-score are ill-defined and being set
to 0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:11
43: UndefinedMetricWarning: Precision and F-score are ill-defined and being set
to 0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)
```

In [227]:
```
cr3
```

Out[227]:
```
'              precision    recall  f1-score   support\n\n           0       0.
00      0.00      0.00       162\n           1       0.40      1.00      0.57
106\n\n   micro avg       0.40      0.40      0.40       268\n   macro avg
0.20      0.50      0.28       268\nweighted avg       0.16      0.40      0.22
268\n'
```

In [228]:
```
from sklearn.metrics import accuracy_score
```

In [229]:
```
score3=accuracy_score(y_test,prediction3)
```

In [230]:
```
score3
```

Out[230]:
```
0.39552238805970147
```

Support Vector Machine (SVM)

In [231]:
```
df.head()
```

Out[231]:

	Ticket	Cabin	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
0	523	63	1	0	3	22.0	1	0	7.2500
1	596	81	2	1	1	38.0	1	0	71.2833
2	669	63	3	1	3	26.0	0	0	7.9250
3	49	55	4	1	1	35.0	1	0	53.1000
4	472	63	5	0	3	35.0	0	0	8.0500

In [232]:
```
x4=df.iloc[:,[0,1,2,4,5,6,7,8]].values
y=df.iloc[:,3].values
```

In [233]:
```
from sklearn.model_selection import train_test_split
```

In [234]:
```
 x4_train, x4_test, y_train, y_test = train_test_split(x4, y, test_size=0.30, ra
```

In [235]:
```
from sklearn.svm import LinearSVC
```

In [236]:
```
svm=LinearSVC()
```

In [237]:
```
svm.fit(x4_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\base.py:931: Convergence
Warning: Liblinear failed to converge, increase the number of iterations.
  "the number of iterations.", ConvergenceWarning)
```

Out[237]:
```
LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True,
     intercept_scaling=1, loss='squared_hinge', max_iter=1000,
     multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
     verbose=0)
```

In [238]:
```
from sklearn.preprocessing import StandardScaler
```

In [239]:
```
sc4=StandardScaler()
```

In [241]:
```
x4_train=sc4.fit_transform(x4_train)
x4_test=sc4.transform(x4_test)
```

In [242]:
```
sc4
```

Out[242]:
```
StandardScaler(copy=True, with_mean=True, with_std=True)
```

```
In [243]: prediction4=svm.predict(x4_test)
```

```
In [244]: prediction4
```

```
Out[244]: array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1], dtype=int64)
```

```
In [245]: from sklearn.metrics import confusion_matrix
```

```
In [246]: cm4=confusion_matrix(y_test,prediction4)
```

```
In [247]: cm4
```

```
Out[247]: array([[  1, 161],
                 [  0, 106]], dtype=int64)
```

```
In [248]: from sklearn.metrics import classification_report
```

```
In [249]: cr4=classification_report(y_test,prediction4)
```

```
In [250]: cr4
```

```
Out[250]: '              precision    recall  f1-score   support\n\n           0       1.
          00      0.01      0.01       162\n           1       0.40      1.00      0.57
          106\n\n   micro avg       0.40      0.40      0.40       268\n   macro avg
          0.70      0.50      0.29       268\nweighted avg       0.76      0.40      0.23
          268\n'
```

```
In [251]: from sklearn.metrics import accuracy_score
```

```
In [252]: score4=accuracy_score(y_test,prediction4)
```

```
In [253]: score4
```

```
Out[253]: 0.39925373134328357
```

Naive Baiyes (GaussianNB)

In [254]:
```
df.head()
```

Out[254]:

	Ticket	Cabin	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
0	523	63	1	0	3	22.0	1	0	7.2500
1	596	81	2	1	1	38.0	1	0	71.2833
2	669	63	3	1	3	26.0	0	0	7.9250
3	49	55	4	1	1	35.0	1	0	53.1000
4	472	63	5	0	3	35.0	0	0	8.0500

In [255]:
```
x5=df.iloc[:,[0,1,2,4,5,6,7,8]].values
y=df.iloc[:,3].values
```

In [256]:
```
from sklearn.model_selection import train_test_split
```

In [257]:
```
x5_train, x5_test, y_train, y_test = train_test_split(x5, y, test_size=0.30, ran
```

In [258]:
```
from sklearn.naive_bayes import GaussianNB
```

In [259]:
```
nb=GaussianNB()
```

In [260]:
```
nb.fit(x5_train,y_train)
```

Out[260]:
```
GaussianNB(priors=None, var_smoothing=1e-09)
```

In [261]:
```
from sklearn.preprocessing import StandardScaler
```

In [262]:
```
sc5=StandardScaler()
```

In [263]:
```
x5_train=sc5.fit_transform(x5_train)
x5_test=sc5.transform(x5_test)
```

In [264]:
```
sc5
```

Out[264]:
```
StandardScaler(copy=True, with_mean=True, with_std=True)
```

In [265]:
```
prediction5=nb.predict(x5_test)
```

```
In [266]: prediction5

Out[266]: array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
                 1, 1, 1, 1], dtype=int64)
```

```
In [267]: from sklearn.metrics import confusion_matrix
```

```
In [268]: cm5=confusion_matrix(y_test,prediction5)
```

```
In [270]: cm5

Out[270]: array([[  4, 158],
                 [  0, 106]], dtype=int64)
```

```
In [271]: from sklearn.metrics import classification_report
```

```
In [272]: cr5=classification_report(y_test,prediction5)
```

```
In [273]: cr5

Out[273]: '              precision    recall  f1-score   support\n\n           0       1.
          00      0.02      0.05       162\n           1       0.40      1.00      0.57
          106\n\n   micro avg       0.41      0.41      0.41       268\n   macro avg
          0.70      0.51      0.31       268\nweighted avg       0.76      0.41      0.26
          268\n'
```

```
In [274]: from sklearn.metrics import accuracy_score
```

```
In [275]: score5=accuracy_score(y_test,prediction5)
```

```
In [276]: score5

Out[276]: 0.41044776119402987
```

```
In [ ]:
```