Assignment 8

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

In [2]:
```
train = pd.read_csv('titanic_train.csv')
```

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

Out[3]:

	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

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

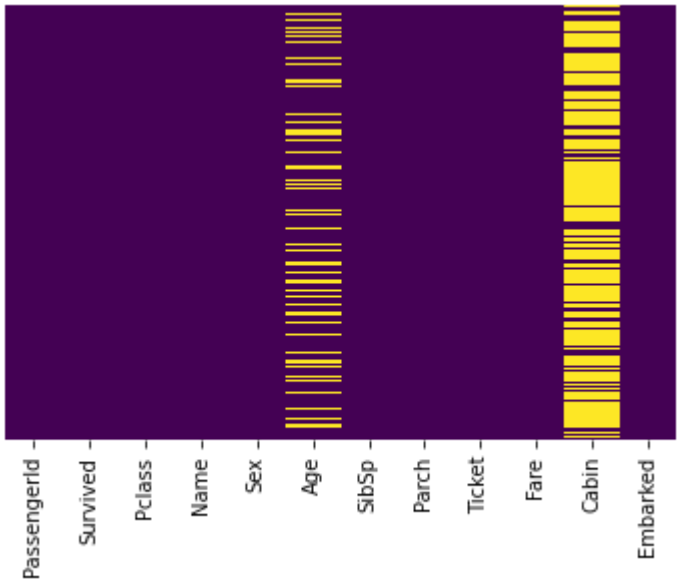
Out[4]:

	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	
...	...	...	...	...	...	...	...	...	...	...	...	
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

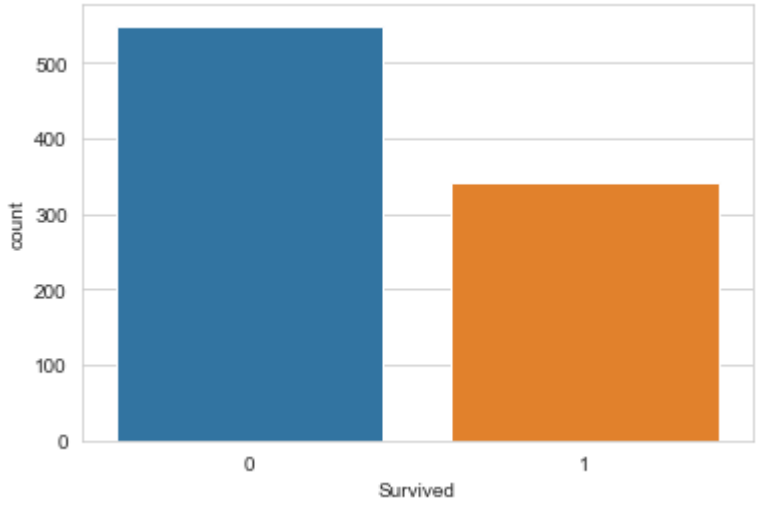
In [5]:
```
sns.heatmap(train.isnull(),yticklabels = False, cbar = False, cmap = 'viridis')
```

Out[5]: <AxesSubplot:>



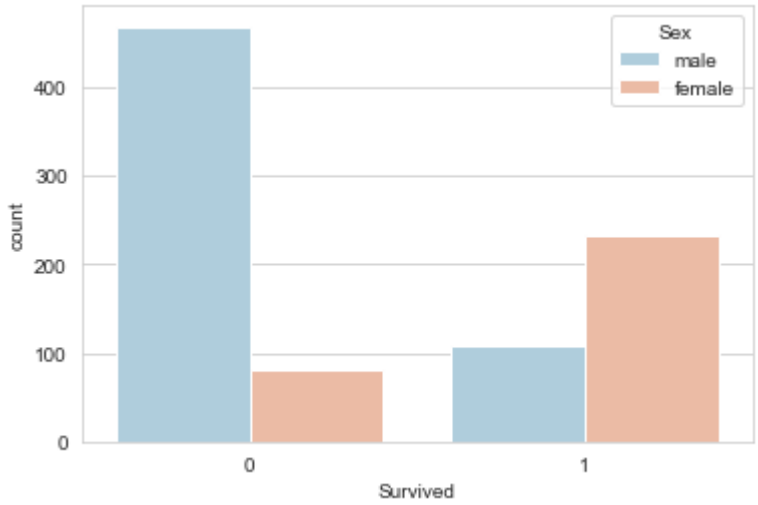
In [6]:
```
sns.set_style('whitegrid')
sns.countplot(x = 'Survived', data = train)
```

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



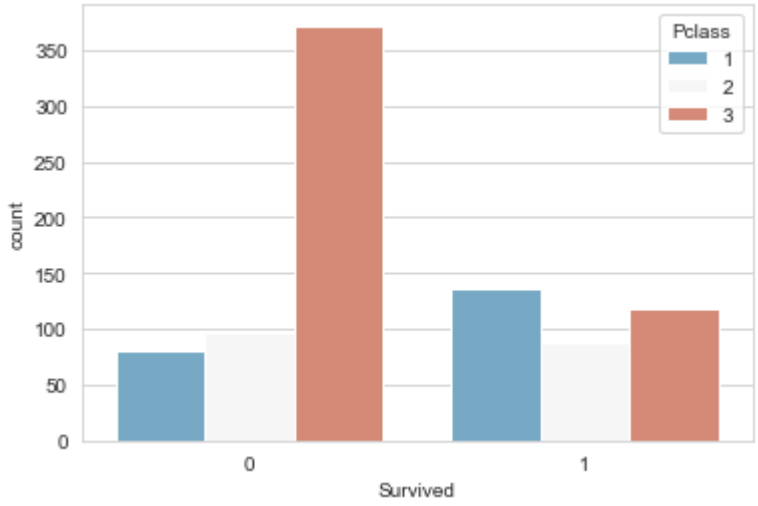
In [7]:
```
sns.set_style('whitegrid')
sns.countplot(x = 'Survived', hue = 'Sex', data = train, palette = "RdBu_r" )
```

Out[7]: <AxesSubplot:xlabel='Survived', ylabel='count'>



In [8]:
```
sns.set_style('whitegrid')
sns.countplot(x = 'Survived', hue = 'Pclass', data = train, palette = "RdBu_r" )
```

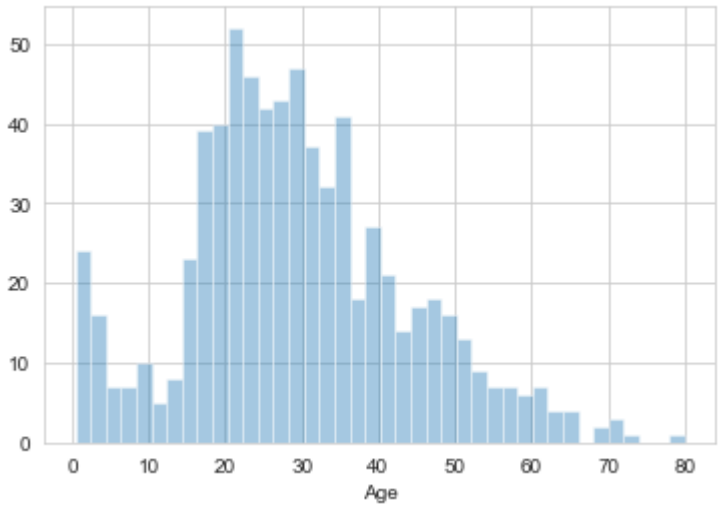
Out[8]: <AxesSubplot:xlabel='Survived', ylabel='count'>



In [9]:
```
sns.distplot(train["Age"].dropna(),kde = False, bins = 40)
```

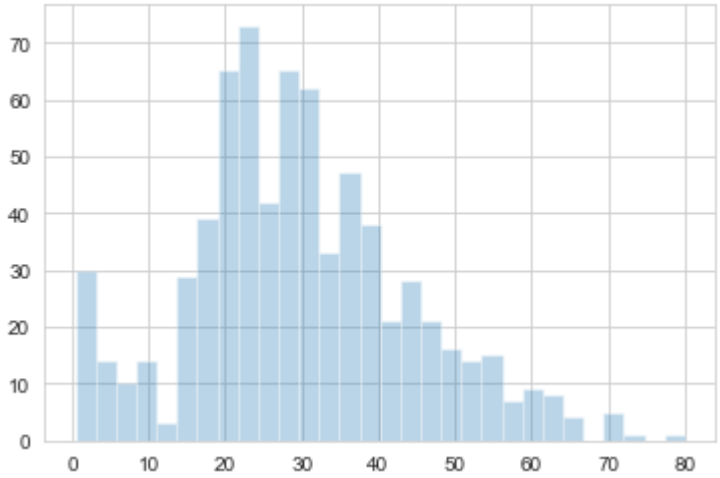
```
C:\Users\DELL\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn
\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and
will be removed in a future version. Please adapt your code to use either `disp
lot` (a figure-level function with similar flexibility) or `histplot` (an axes-
level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[9]: <AxesSubplot:xlabel='Age'>



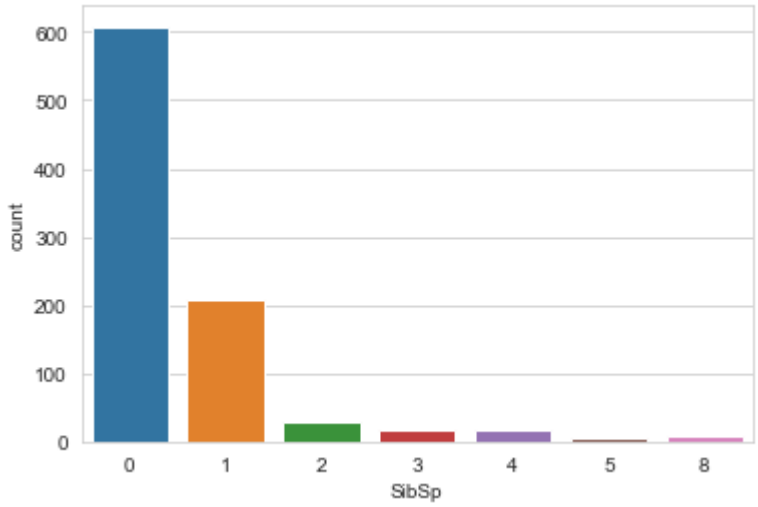
In [10]:
```
train['Age'].hist( bins= 30, alpha = 0.3)
```

Out[10]: <AxesSubplot:>



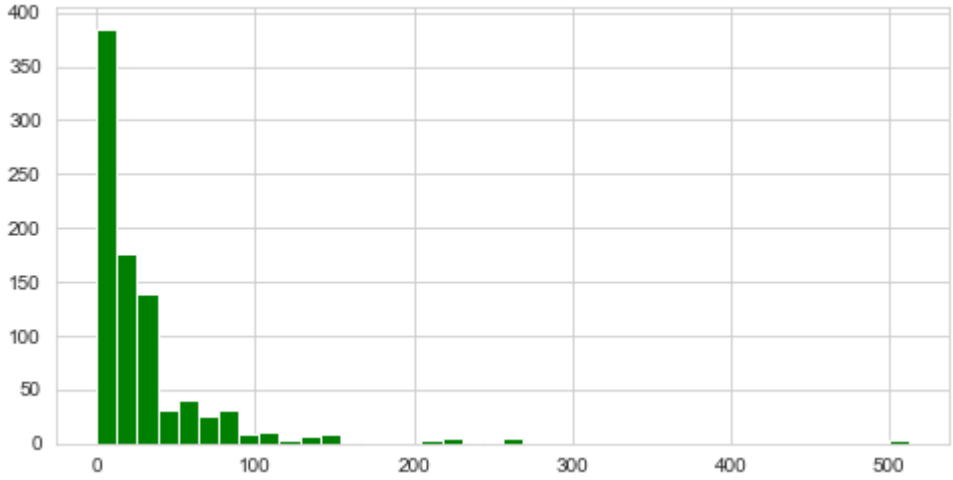
In [11]:
```
sns.countplot(x ='SibSp' , data = train)
```

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



In [12]:
```
train['Fare'].hist(color = 'green', bins = 40, figsize = (8, 4))
```

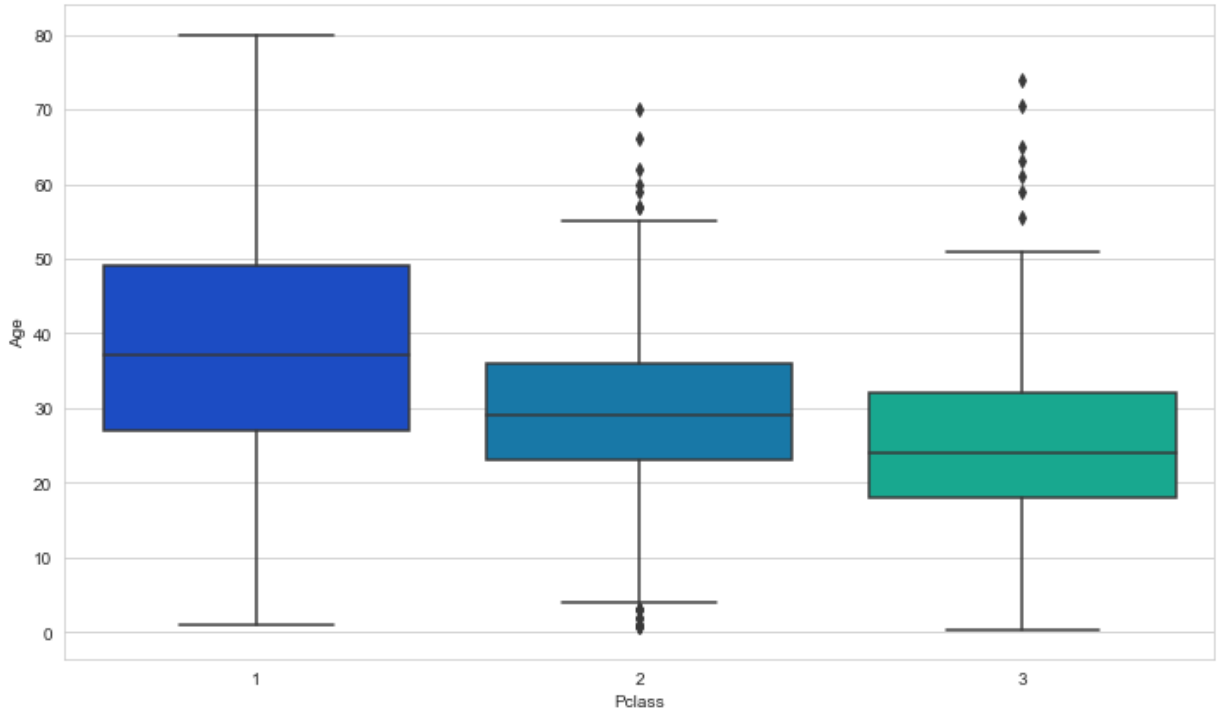
Out[12]: <AxesSubplot:>



In [13]:
```
#Data Cleaning
```

In [14]:
```
plt.figure(figsize = (12, 7))
sns.boxplot(x = 'Pclass', y = 'Age', data = train, palette = 'winter')
```

Out[14]: <AxesSubplot:xlabel='Pclass', ylabel='Age'>

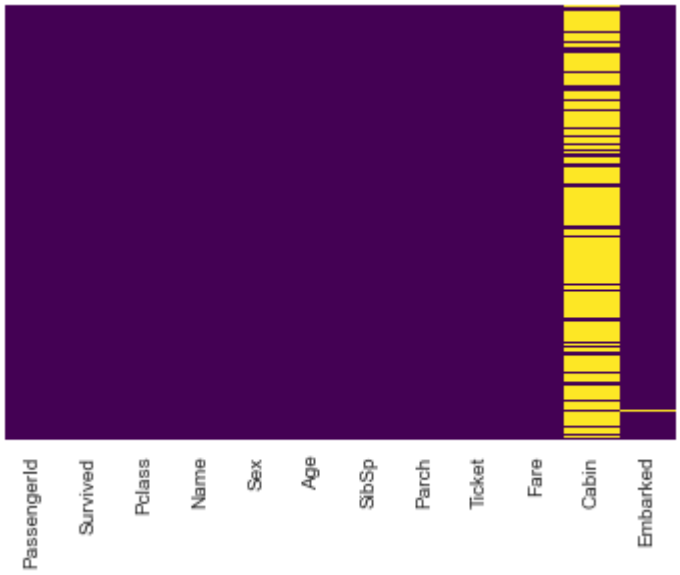


In [15]:
```
def impute_age(cols):
    Age = cols[0]
    Pclass = cols[1]

    if pd.isnull(Age):

        if Pclass == 1:
            return 37

        elif Pclass == 2:
            return 29

        else:
            return 24

    else:
         return Age
```

In [16]:
```
train['Age'] = train[['Age', 'Pclass']].apply(impute_age, axis = 1)
```

In [17]:
```
sns.heatmap(train.isnull(), yticklabels = False, cbar = False, cmap = 'viridis')
```

Out[17]: <AxesSubplot:>



In [18]:
```
train.drop('Cabin', axis = 1, inplace = True)
```

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

Out[19]:

	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	

In [20]:
```
train.dropna(inplace = True)
```

In [21]:
```
train.head()
```

Out[21]:

	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	

In [22]:
```
pd.get_dummies(train ["Embarked"], drop_first = True).head()
```

Out[22]:

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

In [23]:
```
sex = pd.get_dummies(train ["Sex"], drop_first = True)
embark = pd.get_dummies(train['Embarked'], drop_first = True)
```

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

Out[24]:

	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	

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

Out[25]:

	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	

In [26]:
```
train.drop(['Sex', 'Embarked', 'Name', 'Ticket'], axis = 1, inplace = True)
```

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

Out[27]:

	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 [28]:
```
train = pd.concat([train, sex, embark], axis = 1)
```

In [29]:
```
train.head()
```

Out[29]:

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

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

Out[30]:

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

In [31]:
```
train['Survived'].head()
```

Out[31]:
```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

In [32]:
```
from sklearn.model_selection import train_test_split
```

In [33]:
```
X_train, X_test, y_train, y_test = train_test_split(train.drop('Survived', axis =
                                                    train['Survived'], test_size
```

In [34]:
```
from sklearn.linear_model import LogisticRegression
```

In [35]:
```
logmodel = LogisticRegression()
logmodel.fit(X_train, y_train)
```

```
C:\Users\DELL\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn
\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs failed to converge (s
tatus=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-
learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regressi
on (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regressi
on)
  n_iter_i = _check_optimize_result(
```

Out[35]:
```
LogisticRegression()
```

In [36]:
```
predictions = logmodel.predict(X_test)
```

In [37]:
```
from sklearn.metrics import confusion_matrix
```

In [38]:
```
accuracy=confusion_matrix(y_test,predictions)
```

In [39]:
```
accuracy
```

Out[39]:
```
array([[141,  25],
       [ 24,  77]], dtype=int64)
```

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

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

Out[41]:
```
0.8164794007490637
```

In [42]:
```
predictions
```

Out[42]:
```
array([1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0,
       0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0,
       0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1,
       0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0,
       0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0,
       0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 0, 0], dtype=int64)
```

In [43]:
```
from sklearn.metrics import classification_report
```

In [44]:
```
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       0.85      0.85      0.85       166
           1       0.75      0.76      0.76       101

    accuracy                           0.82       267
   macro avg       0.80      0.81      0.81       267
weighted avg       0.82      0.82      0.82       267
```