```
In [ ]: Task-o1: Titanic Classification (Data Science Intern)
```

Import Libraries

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

The Dataset

```python
train = pd.read_csv('titanic_classication.csv')
```

```python
train.head()
```

	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

Exploratory Data Analysis

Missing Data

we can use seaborn to create a simple heatmap to see where we are missing data!

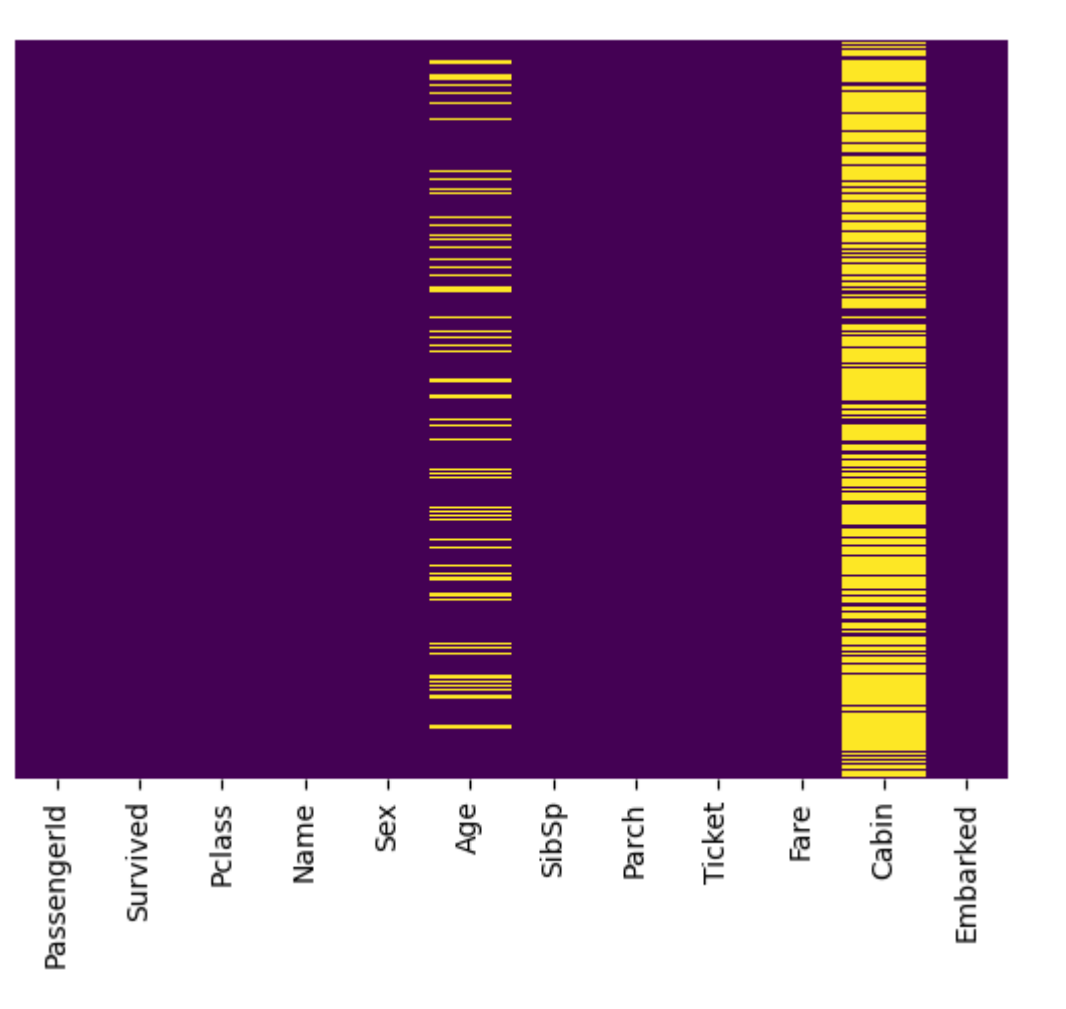
```python
train.isnull()
```

	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
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

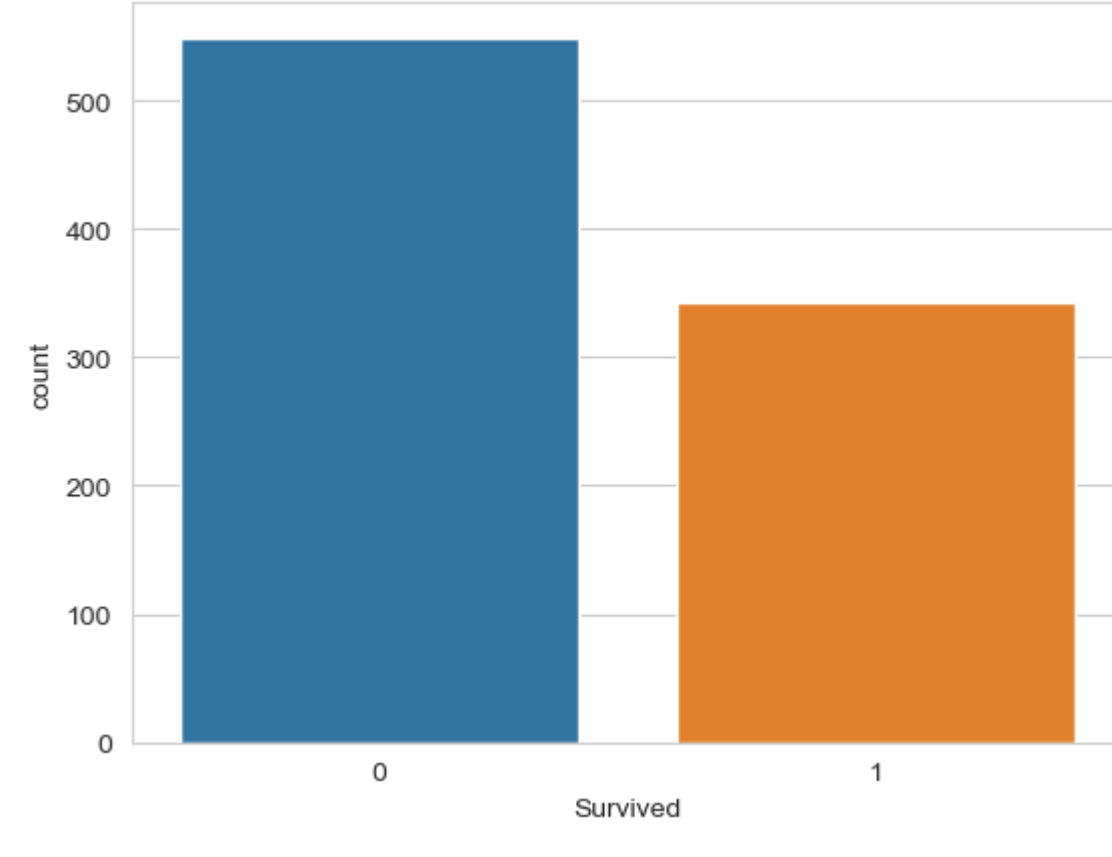
```python
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

<Axes: >



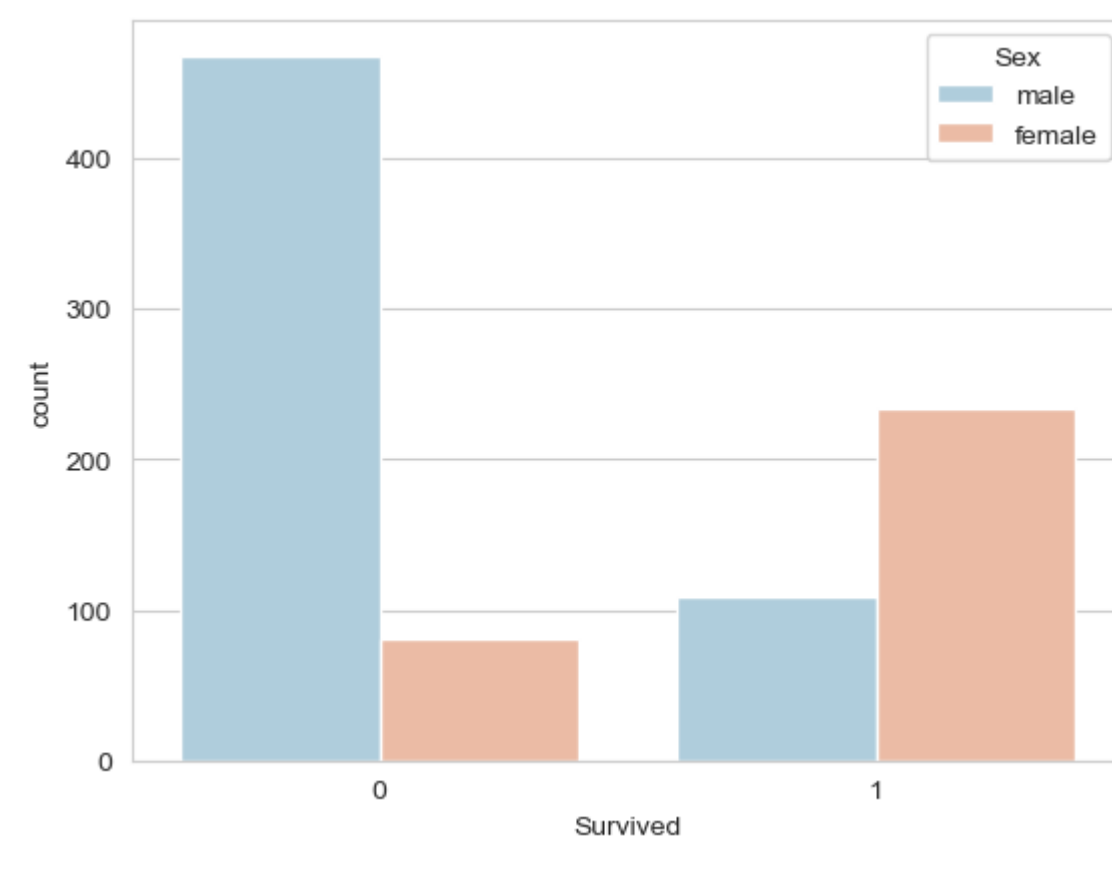

```python
sns.set_style('whitegrid')
sns.countplot(x='Survived',data=train)
```

<Axes: xlabel='Survived', ylabel='count'>



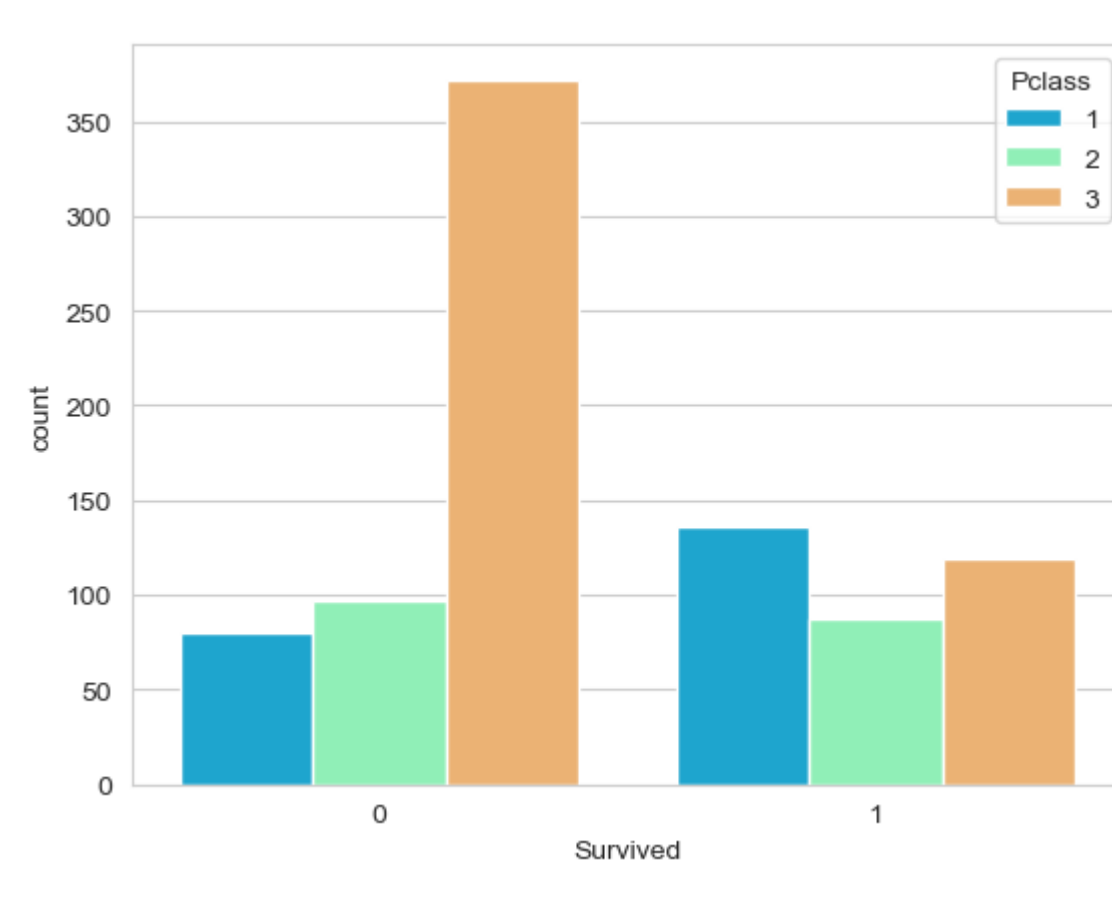

```python
sns.set_style('whitegrid')
sns.countplot(x='Survived',hue='Sex',data=train,palette='RdBu_r')
```

<Axes: xlabel='Survived', ylabel='count'>




```python
sns.set_style('whitegrid')
sns.countplot(x='Survived',hue='Pclass',data=train,palette='rainbow')
```

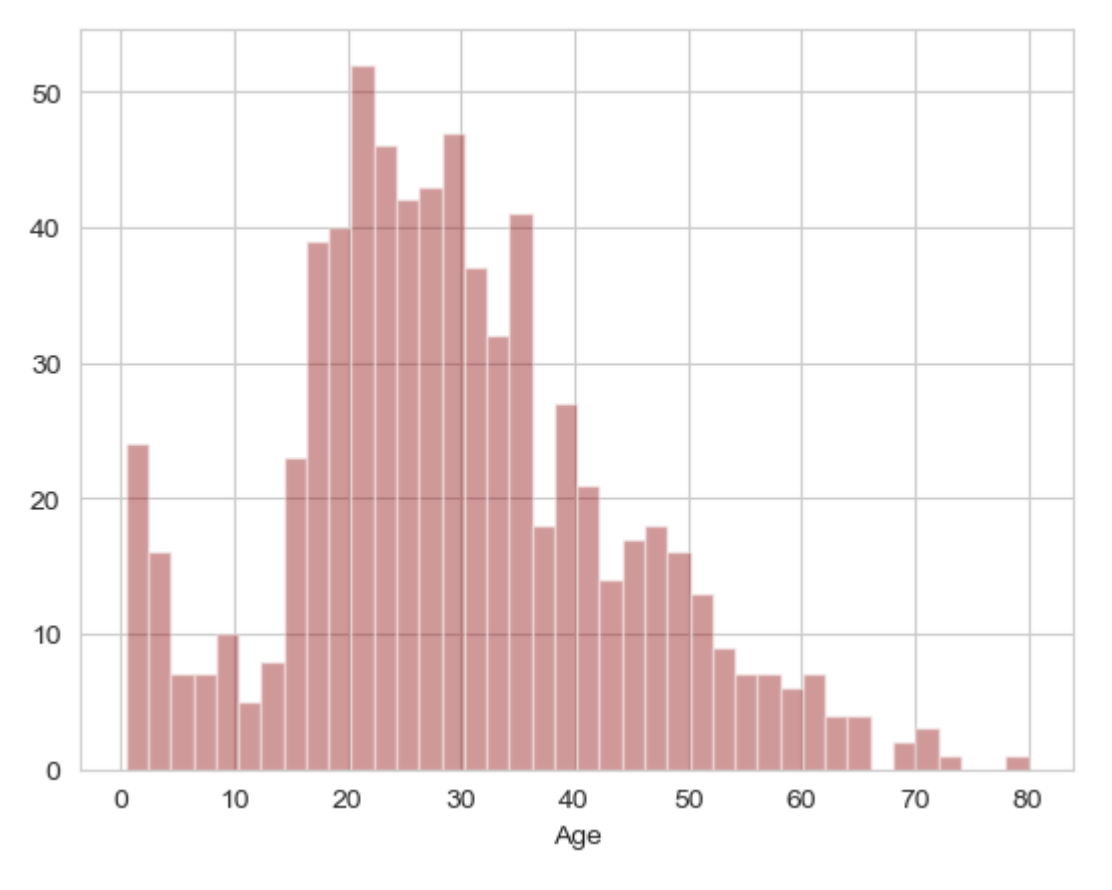
<Axes: xlabel='Survived', ylabel='count'>




```python
sns.distplot(train['Age'].dropna(),kde=False,color='darkred',bins=40)
```

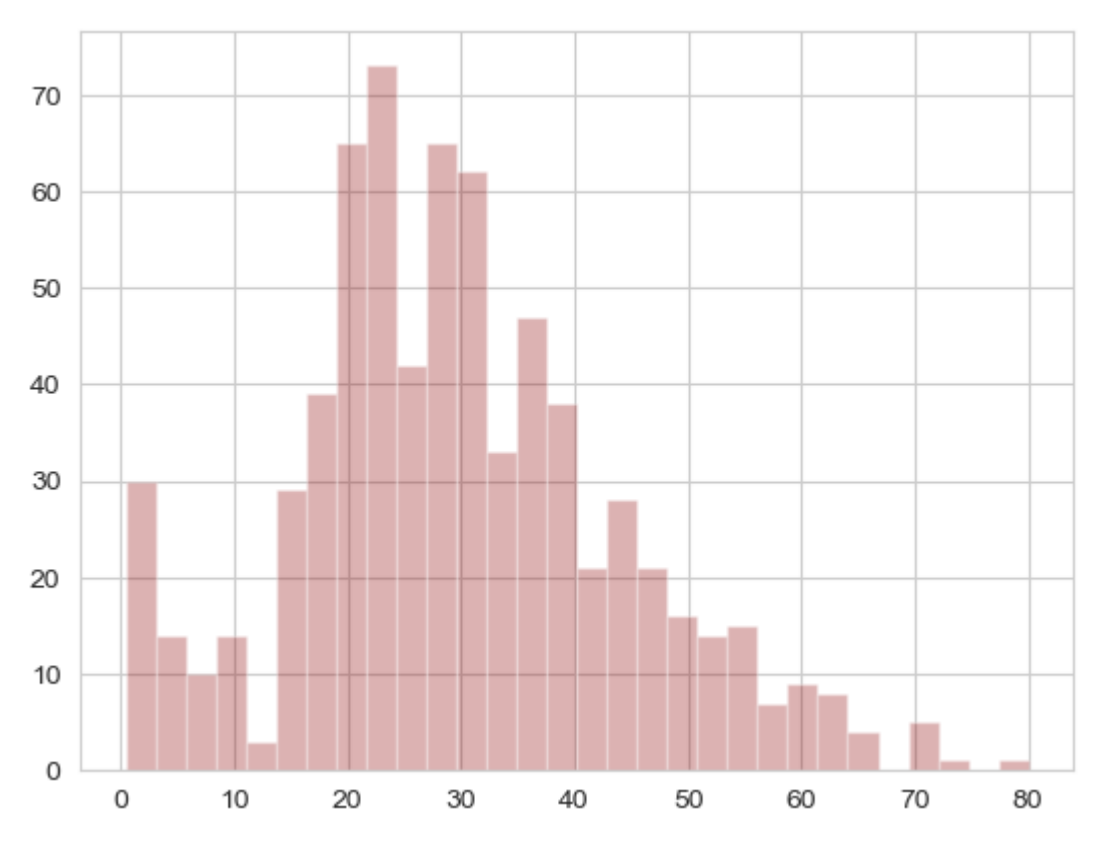
```
C:\Users\INDIA\AppData\Local\Temp\ipykernel_17136\2002818437.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(train['Age'].dropna(),kde=False,color='darkred',bins=40)
```

<Axes: xlabel='Age'>



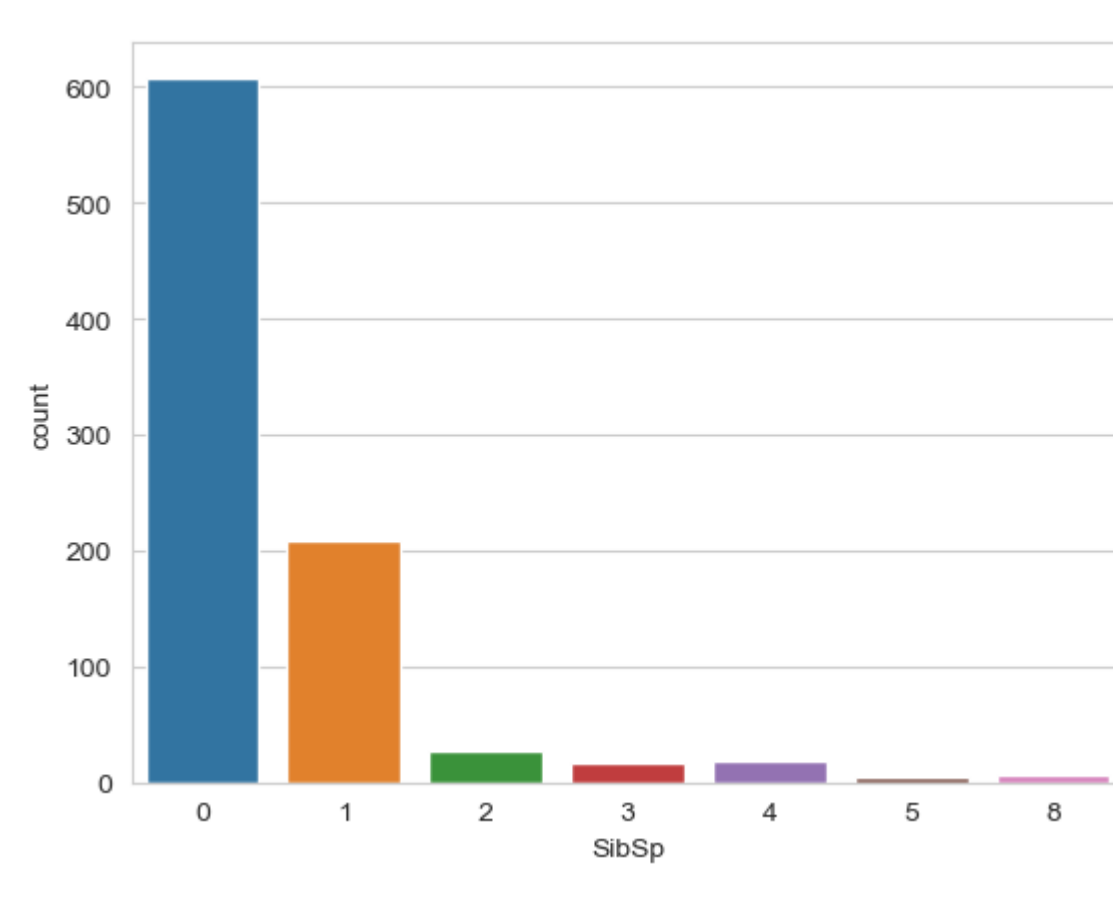

```python
train['Age'].hist(bins=30,color='darkred',alpha=0.3)
```

<Axes: >



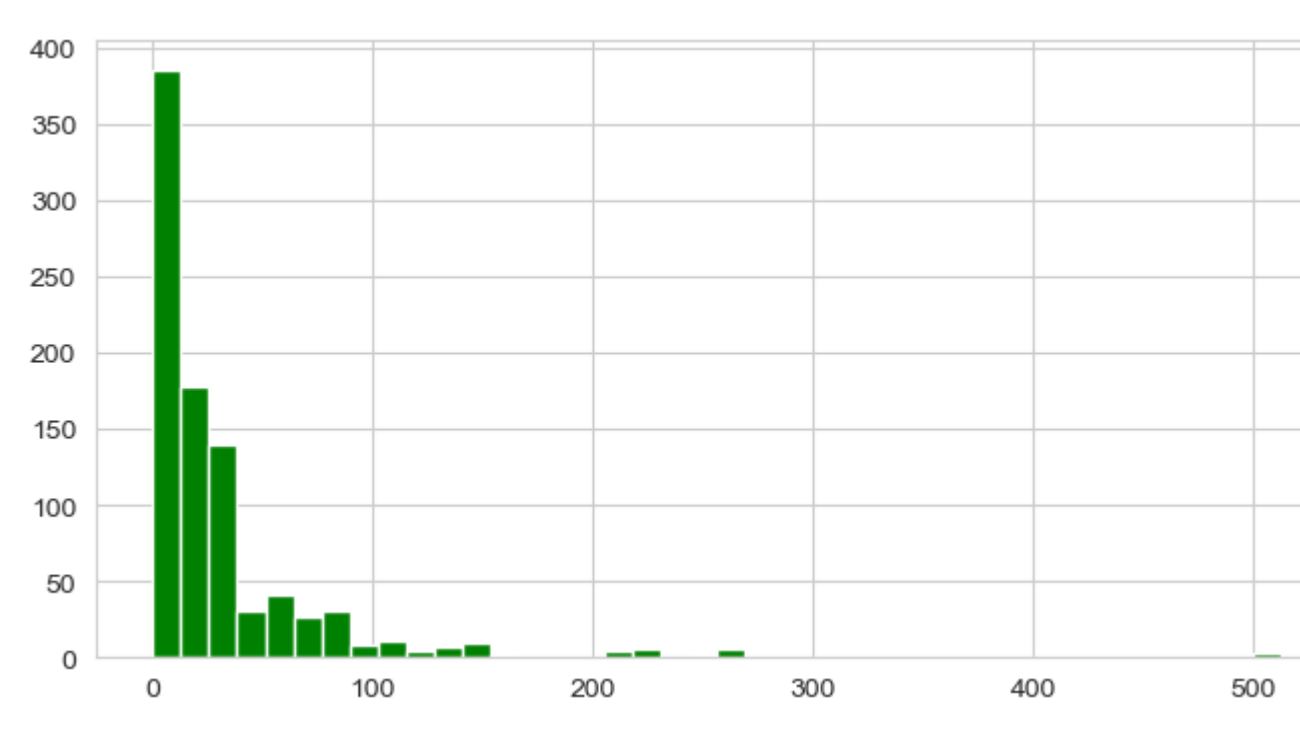
```python
sns.countplot(x='SibSp',data=train)
```

<Axes: xlabel='SibSp', ylabel='count'>




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

<Axes: >



Cufflinks for plots Let's take a quick moment to show an example of cufflinks!

```python
import cufflinks as cf
cf.go_offline()
```

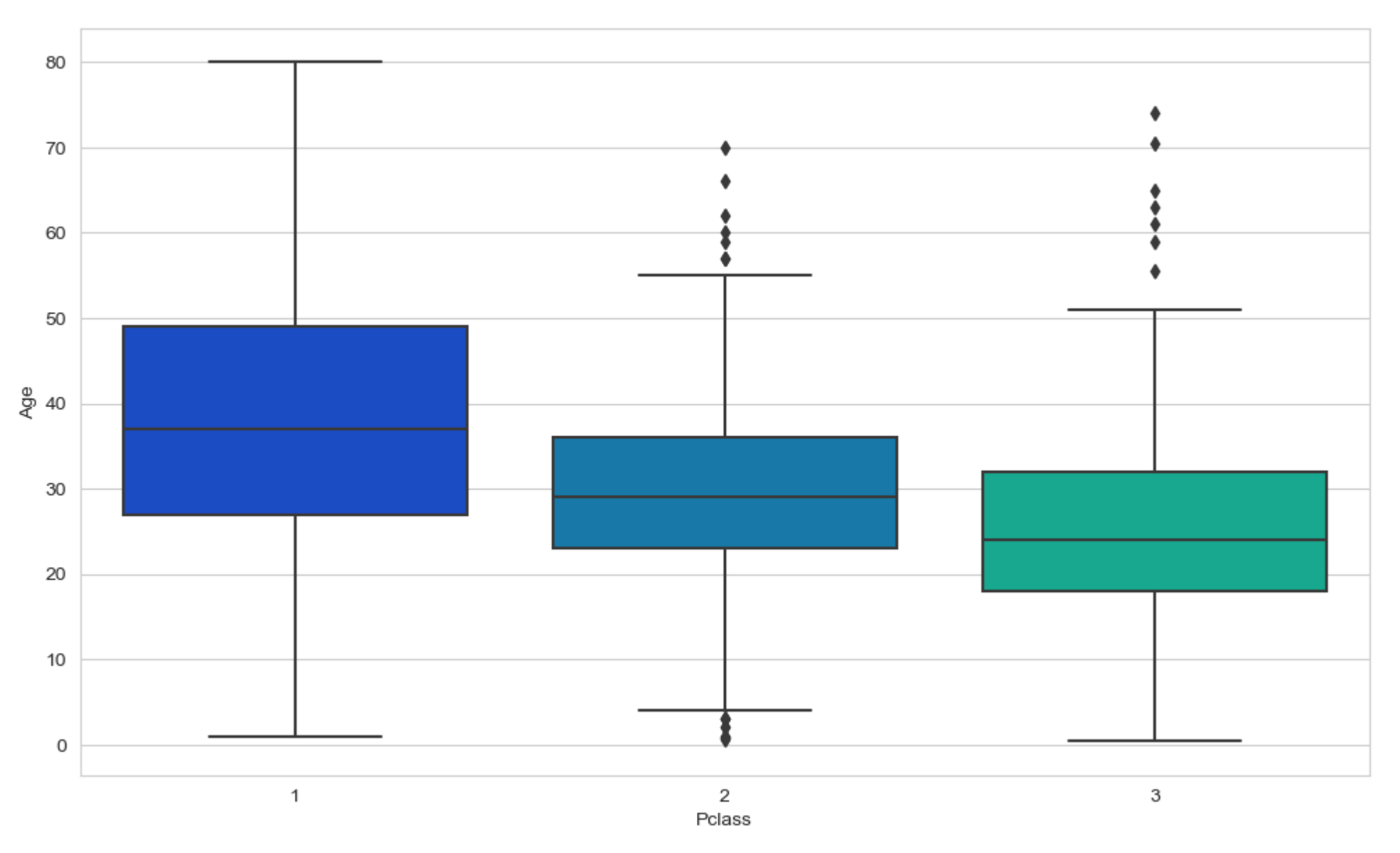
```python
train['Fare'].iplot(kind='hist',bins=30,color='green')
```

Data Cleaning

We want to fill in missing age data instead of just dropping the missing age data rows. One way to do this is by filling in the mean age of all the passengers (imputation). However we can be smarter about this and check the average age by passenger class. For example:

```python
plt.figure(figsize=(12, 7))
sns.boxplot(x='Pclass',y='Age',data=train,palette='winter')
```

<Axes: xlabel='Pclass', ylabel='Age'>




```python
#We can see the wealthier passengers in the higher classes tend to be older, which makes sense. We'll use these average age values to impute based on Pclass fo
```

```python
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
```

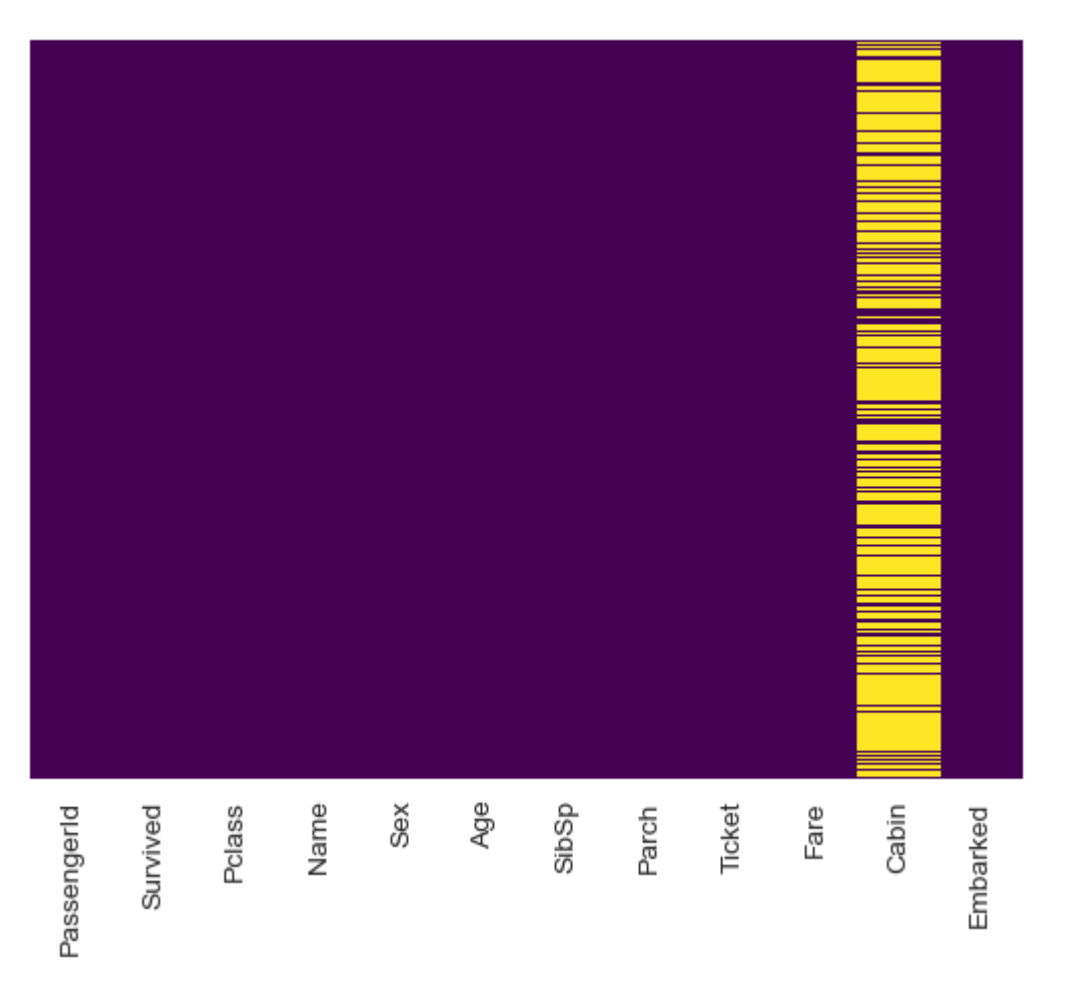
```
Now apply that funtion
```

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

Now lets Check that heat map agin!

```python
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

<Axes: >



ahead and drop the Cabin column and the row in Embarked that is NaN

```python
train.drop('cabin',axis=1,inplace=True)
```

```python
train.head()
```

	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

```python
tain.dropna(inplace=True)
```

Converting Categorical Features

Convert categorical features to dummy variables using pandas! Otherwise our machine learning algorithm won't be able to directly take in those features as inputs.

```python
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          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  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

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

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

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

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

```python
train.head()
```

	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

```python
train = pd.connect([train,sex,embark],axis=1)
```

```python
train.head()
```

	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

#Building A Logistic Regression

```
Train Test Split
```

```python
train.drop('Survived',axis=1).head()
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```python
train['Survived'].head()
```

```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

```python
from sklearn.model_selection import train_test_split
```

```python
x_train,x_test.y_train,y_test = train_test_split(train_split(train.drop)'Survived',axis=1),train['Survived'], test_size=0.30,random_state=101)
```

Training and Predicting

```python
from sklearn.linear_model import LogisticRegression
```

```python
predictions = logmodel.predict(X_test)
```

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

```python
accuracy=confusion_matrix(y_test,predictions)
```

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

```python
accuracy=accuracy_score(y_test,predictions)
accuracy
```

Evalution

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

```python
print(classification_report(y_test,predictions))
```

precision recall f1-score support

```
          0       0.81      0.93      0.86       163
          1       0.85      0.65      0.74       104
```

avg / total 0.82 0.82 0.81 267

```
                    ***End Of The Code***
```

```
                   ********THANK YOU**********
```