In [103…
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [105…
```
tytanictrain=pd.read_csv('tytanicsurvive.csv')
```

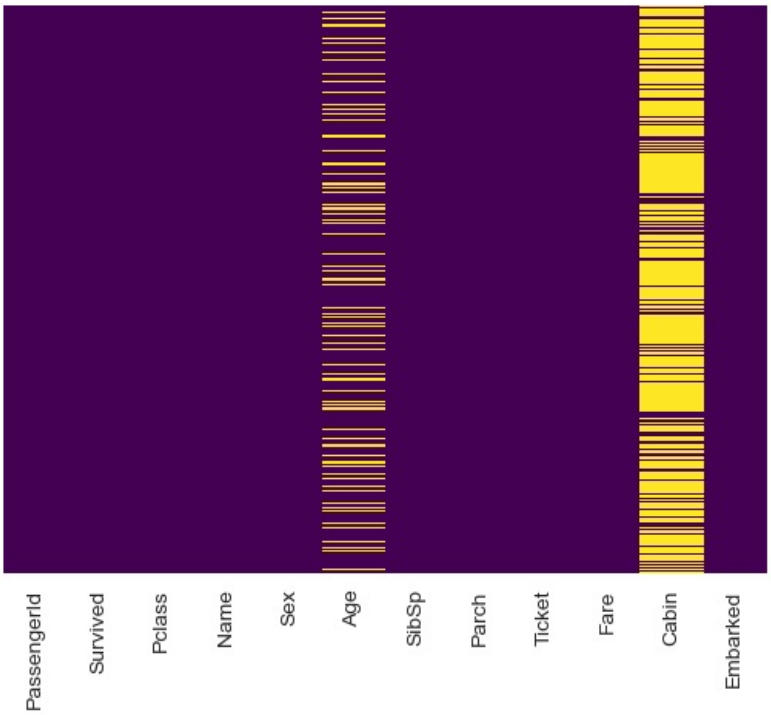
In [107…
```
tytanictrain
```

Out[107…

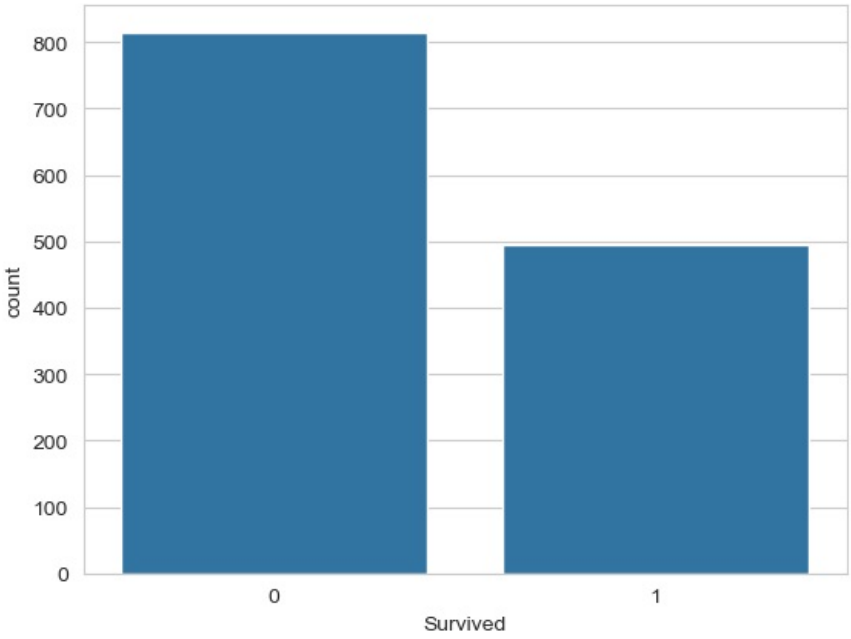
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 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
...	...	...	...	...	...	...	...	...	...	...	...	...
1304	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
1305	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
1306	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
1307	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
1308	1309	0	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

1309 rows × 12 columns

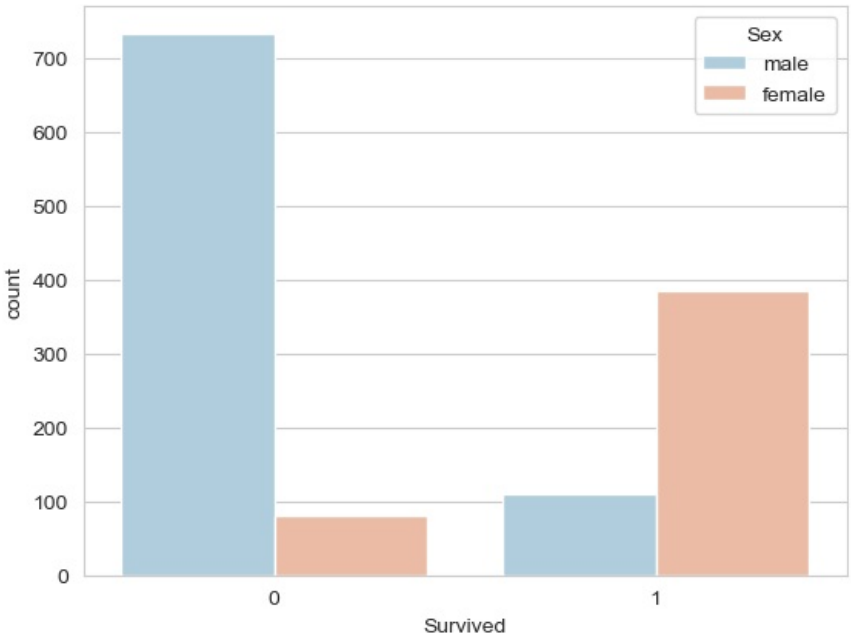
In [109…
```
sns.heatmap(tytanictrain.isnull(),yticklabels=False,cbar=False,cmap='viridis')
plt.show()
```



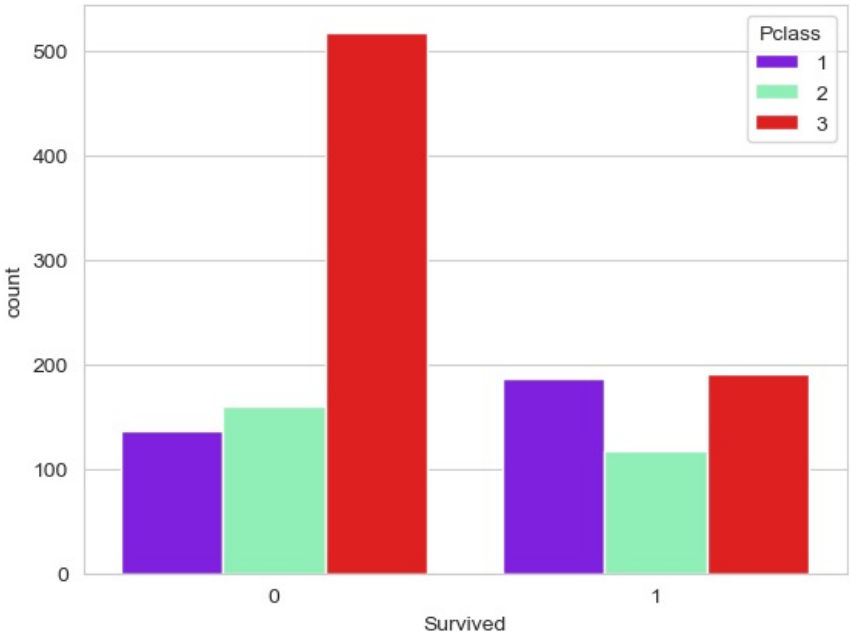
In [110…
```
sns.set_style('whitegrid')
sns.countplot(x='Survived',data=tytanictrain)
plt.show()
```



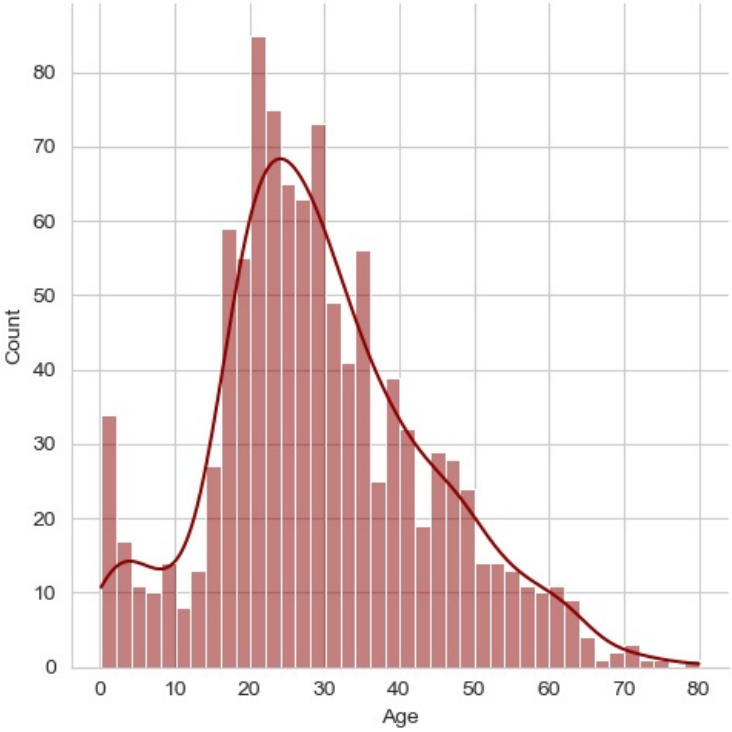
In [113…
```
sns.set_style('whitegrid')
sns.countplot(x='Survived',hue='Sex',data=tytanictrain,palette='RdBu_r')
plt.show()
```



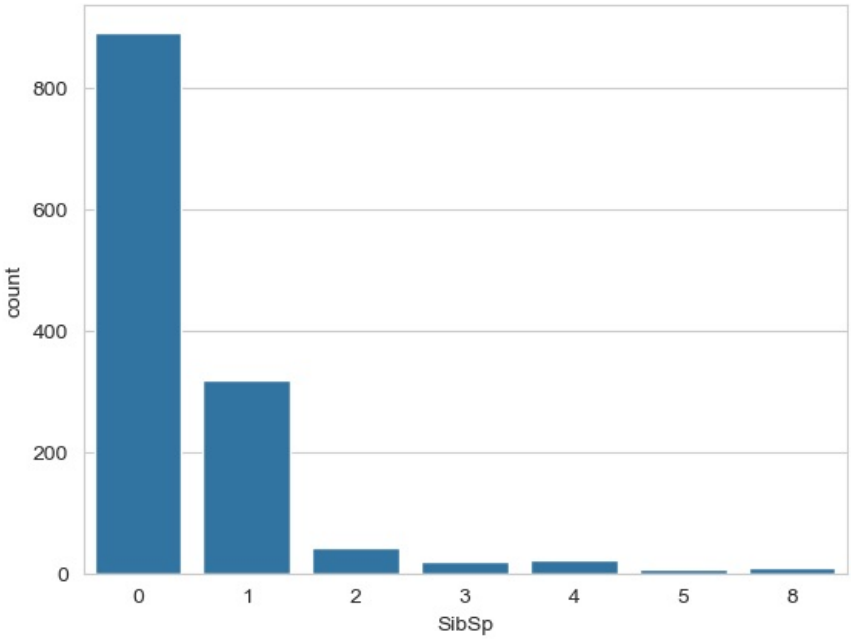
In [115…
```
sns.set_style('whitegrid')
sns.countplot(x='Survived',hue='Pclass',data=tytanictrain,palette='rainbow')
plt.show()
```



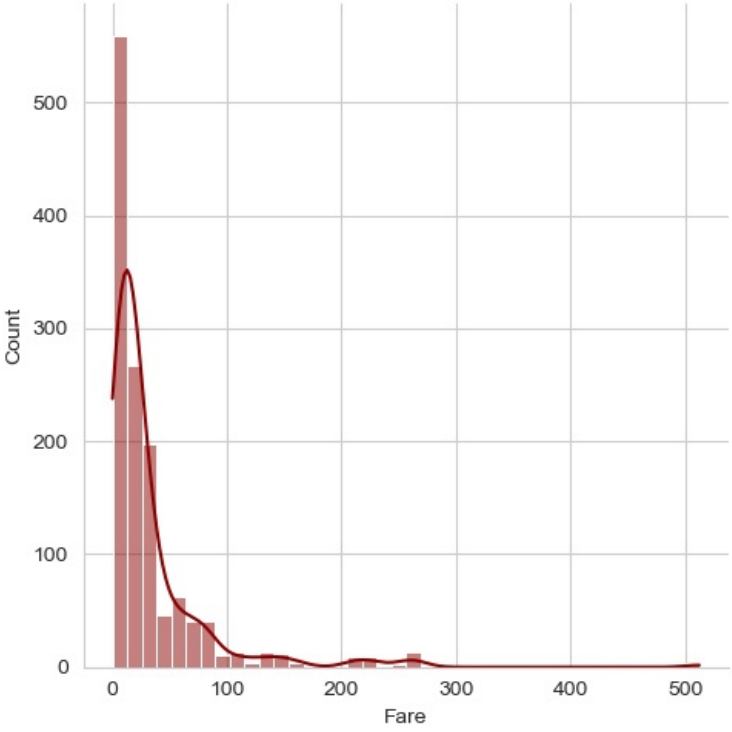
In [117…
```
sns.displot(tytanictrain['Age'].dropna(),kde=True,color='darkred',bins=40)
plt.show()
```



In [118…
```
sns.countplot(x='SibSp',data=tytanictrain)
plt.show()
```

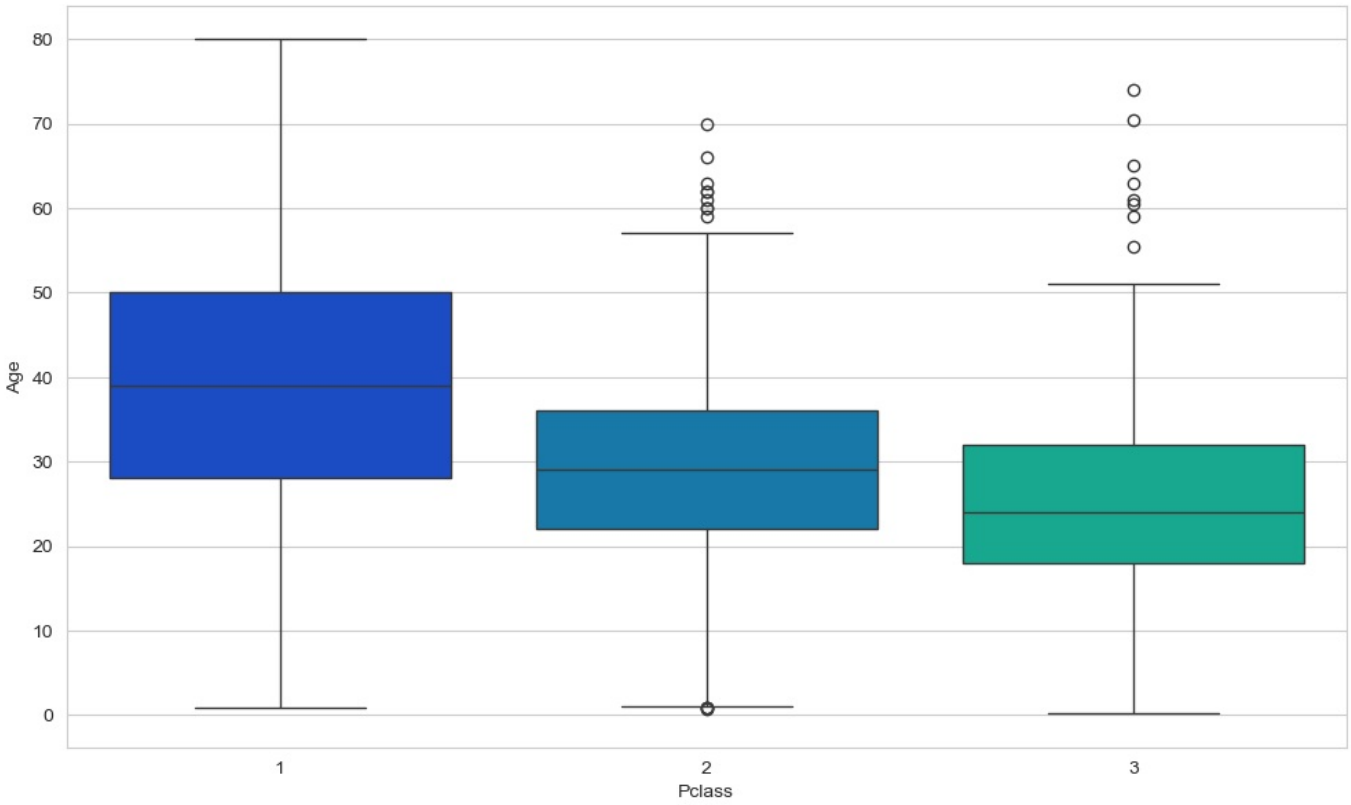


In [119…
```
sns.displot(tytanictrain['Fare'].dropna(),kde=True,color='darkred',bins=40)
plt.show()
```



In [120…
```
plt.figure(figsize=(12,7))
sns.boxplot(x='Pclass',y='Age',data=tytanictrain,palette='winter')
plt.show()
```

```
C:\Users\EVERLYN\AppData\Local\Temp\ipykernel_11652\812286760.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.boxplot(x='Pclass',y='Age',data=tytanictrain,palette='winter')
```

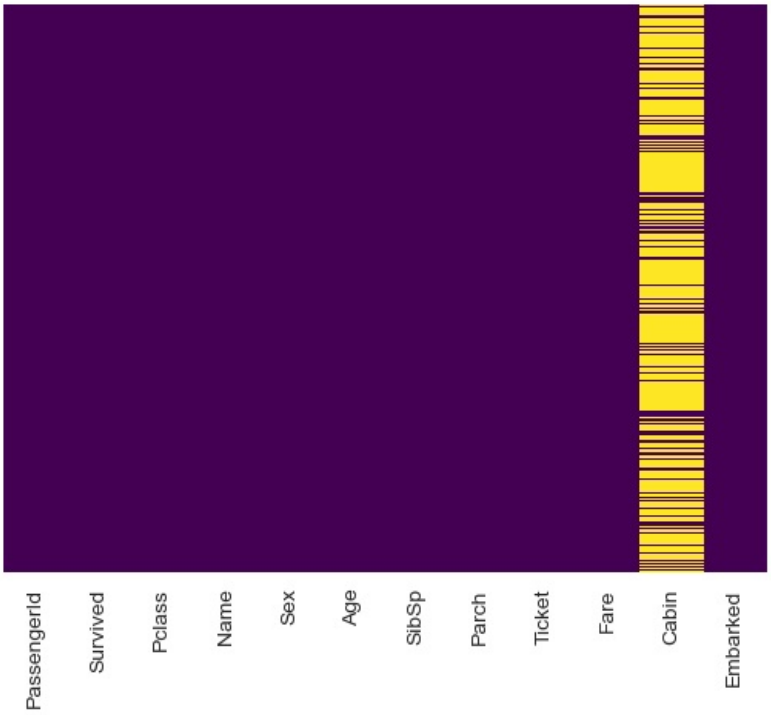


In [122…
```
def inputeage(cols):
    Age=cols[0]
    Pclass=cols[1]

    if pd.isnull(Age):
        if Pclass==1:
            return 39

        elif Pclass==2:
            return 29

        else:
            return 24

    else:
        return Age
```

In [124…
```
tytanictrain['Age']=tytanictrain[['Age','Pclass']].apply(inputeage,axis=1)
```

```
C:\Users\EVERLYN\AppData\Local\Temp\ipykernel_11652\1835021258.py:2: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value by position, use `ser.iloc[pos]`
  Age=cols[0]
C:\Users\EVERLYN\AppData\Local\Temp\ipykernel_11652\1835021258.py:3: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value by position, use `ser.iloc[pos]`
  Pclass=cols[1]
```

In [127…
```
sns.heatmap(tytanictrain.isnull(),yticklabels=False,cbar=False,cmap='viridis')
plt.show()
```



In [131…
```
tytanictrain.drop('Cabin',axis=1,inplace=True)
```

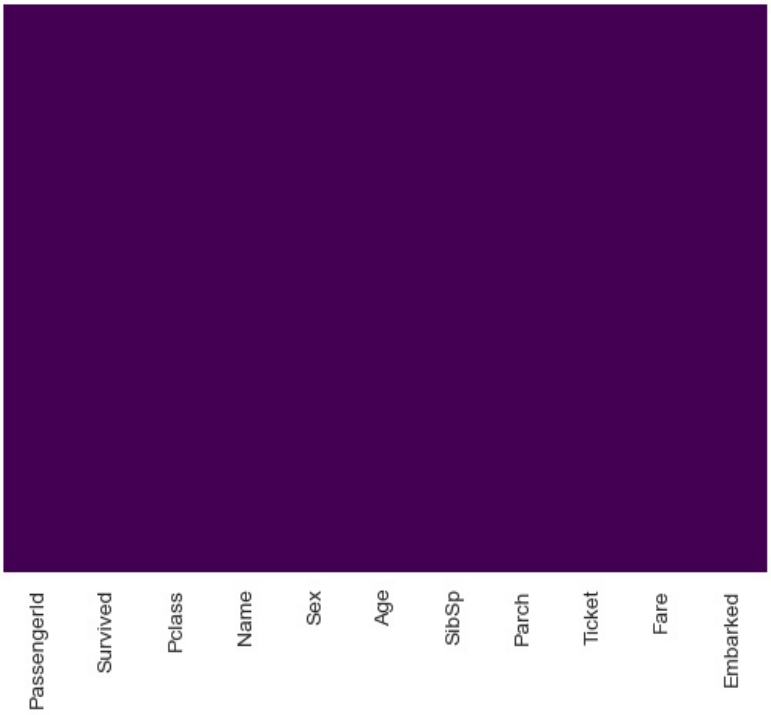
In [133…
```
tytanictrain
```

Out[133…

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	S
...	...	...	...	...	...	...	...	...	...	...	...
1304	1305	0	3	Spector, Mr. Woolf	male	24.0	0	0	A.5. 3236	8.0500	S
1305	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C
1306	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	S
1307	1308	0	3	Ware, Mr. Frederick	male	24.0	0	0	359309	8.0500	S
1308	1309	0	3	Peter, Master. Michael J	male	24.0	1	1	2668	22.3583	C

1309 rows × 11 columns

In [135…
```
sns.heatmap(tytanictrain.isnull(),yticklabels=False,cbar=False,cmap='viridis')
plt.show()
```



In [137…
```
sex=pd.get_dummies(tytanictrain['Sex'],drop_first=True)
embarked=pd.get_dummies(tytanictrain['Embarked'],drop_first=True)
tytanictrain.drop(['Sex','Embarked','Name','Ticket'],axis=1,inplace=True)
tytanictrain=pd.concat([tytanictrain,sex,embarked],axis=1)
tytanictrain
```

Out[137…

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	male	Q	S
0	1	0	3	22.0	1	0	7.2500	True	False	True
1	2	1	1	38.0	1	0	71.2833	False	False	False
2	3	1	3	26.0	0	0	7.9250	False	False	True
3	4	1	1	35.0	1	0	53.1000	False	False	True
4	5	0	3	35.0	0	0	8.0500	True	False	True
...	...	...	...	...	...	...	...	...	...	...
1304	1305	0	3	24.0	0	0	8.0500	True	False	True
1305	1306	1	1	39.0	0	0	108.9000	False	False	False
1306	1307	0	3	38.5	0	0	7.2500	True	False	True
1307	1308	0	3	24.0	0	0	8.0500	True	False	True
1308	1309	0	3	24.0	1	1	22.3583	True	False	False

1309 rows × 10 columns

In [139…
```
tytanictrain['male']=tytanictrain['male'].astype(int)
tytanictrain['Q']=tytanictrain['Q'].astype(int)
tytanictrain['S']=tytanictrain['S'].astype(int)
```

In [141…
```
tytanictrain
```

Out[141…

	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
...	...	...	...	...	...	...	...	...	...	...
1304	1305	0	3	24.0	0	0	8.0500	1	0	1
1305	1306	1	1	39.0	0	0	108.9000	0	0	0
1306	1307	0	3	38.5	0	0	7.2500	1	0	1
1307	1308	0	3	24.0	0	0	8.0500	1	0	1
1308	1309	0	3	24.0	1	1	22.3583	1	0	0

1309 rows × 10 columns

In [195…
```
tytanictrain=tytanictrain.dropna()
```

In [197…
```
tytanictrain
```

Out[197…

	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
...	...	...	...	...	...	...	...	...	...	...
1304	1305	0	3	24.0	0	0	8.0500	1	0	1
1305	1306	1	1	39.0	0	0	108.9000	0	0	0
1306	1307	0	3	38.5	0	0	7.2500	1	0	1
1307	1308	0	3	24.0	0	0	8.0500	1	0	1
1308	1309	0	3	24.0	1	1	22.3583	1	0	0

1308 rows × 10 columns

In [199…
```
tytanictrain.drop('Survived',axis=1)
```

Out[199…

	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
...	...	...	...	...	...	...	...	...	...
1304	1305	3	24.0	0	0	8.0500	1	0	1
1305	1306	1	39.0	0	0	108.9000	0	0	0
1306	1307	3	38.5	0	0	7.2500	1	0	1
1307	1308	3	24.0	0	0	8.0500	1	0	1
1308	1309	3	24.0	1	1	22.3583	1	0	0

1308 rows × 9 columns

In [201…
```
tytanictrain['Survived']
```

Out[201…
```
0       0
1       1
2       1
3       1
4       0
       ..
1304    0
1305    1
1306    0
1307    0
1308    0
Name: Survived, Length: 1308, dtype: int64
```

In [203…
```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,classification_report
```

In [205…
```
X=tytanictrain.drop('Survived',axis=1)
```

In [207…
```
y=tytanictrain['Survived']
```

In [209…
```
X_train, X_test, y_train, y_test=train_test_split(X,y,test_size=0.3,random_state=101)
```

In [211…
```
model=LogisticRegression()
```

In [213…
```
model.fit(X_train, y_train)
```

```
C:\Users\EVERLYN\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:469: ConvergenceWarning: lbfgs fa
iled to converge (status=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
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Out[213…
```
LogisticRegression
LogisticRegression()
```

In [215…
```
y_pred=model.predict(X_test)
```

In [219…
```
accuracy= accuracy_score(y_test,y_pred)
report=classification_report(y_test,y_pred)
```

In [221…
```
print("ACCURACY:",accuracy)
print("CLASSIFICATION REPORT:\n", report)
```

```
ACCURACY: 0.8727735368956743
CLASSIFICATION REPORT:
              precision    recall  f1-score   support

           0       0.87      0.93      0.90       242
           1       0.87      0.78      0.83       151

    accuracy                           0.87       393
   macro avg       0.87      0.86      0.86       393
weighted avg       0.87      0.87      0.87       393
```

In [225…
```
predictions=y_pred
```

In [227…
```
print("PREDICTIONS:",predictions)
```

```
PREDICTIONS: [0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 1 1 1 0 1 1 0 0 1 0 0 1 0 1 1 0 0
 1 1 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 1 1 0 1 0 0 0 1 0 1 0 0 0 0 0 1 1
 1 0 0 1 0 1 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 1 0 0
 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 1 0
 0 1 1 0 1 0 0 0 0 0 0 0 0 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 0 0 0 1 1 1 0 0 0 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 1 0
 0 1 0 0 1 1 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 1 1
 0 0 0 0 0 0 0 1 1 0 1 0 0 1 1 0 1 0 1 0 0 1 1 0 1 0 0 1 0 0 1 1 1 0 0 1 0
 1 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 1 0 0 0 0 1
 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 1 1 0 0 0 1 1 1 1 0 1 1 0 0 0 0 0 1 1 0
 0 0 0 0 0 1 0 0 1 0 0 0 1 1 1 0 1 1 0 0 0 0 0 0]
```