In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
import warnings
warnings.simplefilter(action='ignore')
sns.set_theme(style="darkgrid",palette=sns.color_palette("muted"))
```

In [2]:
```
data = pd.read_csv("Titanic-Dataset.csv")
```

In [3]:
```
data.head(10)
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
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
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708

In [4]:
```
data.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
```

In [5]:
```
data.duplicated().sum()
```

Out[5]:
```
0
```

In [6]:
```
data.isna().sum()
```

Out[6]:
```
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 [7]:
```
data['Embarked'] = data['Embarked'].replace({'S':'Southampton','Q':'Queenst
```

In [8]:
```
data = data.drop(columns=["PassengerId","Name","Cabin","Ticket"])
```

In [9]:
```
data[data["SibSp"]==8]
```

Out[9]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
159	0	3	male	NaN	8	2	69.55	Southampton
180	0	3	female	NaN	8	2	69.55	Southampton
201	0	3	male	NaN	8	2	69.55	Southampton
324	0	3	male	NaN	8	2	69.55	Southampton
792	0	3	female	NaN	8	2	69.55	Southampton
846	0	3	male	NaN	8	2	69.55	Southampton
863	0	3	female	NaN	8	2	69.55	Southampton

In [10]:
```
data = data.drop(data[data["SibSp"]==8].index)
```

In [11]:
```
data.groupby(["Survived","Pclass","SibSp"])["Age"].mean()
```

Out[11]:

```
Survived  Pclass  SibSp
0         1       0        46.375000
                  1        38.500000
                  2        44.000000
                  3        19.000000
          2       0        33.833333
                  1        34.239130
                  2        25.000000
          3       0        29.103175
                  1        27.966667
                  2        24.444444
                  3         5.428571
                  4         6.800000
                  5        10.200000
1         1       0        34.594203
                  1        36.998298
                  2        35.500000
                  3        23.500000
          2       0        29.333261
                  1        22.421875
                  2        13.250000
                  3        30.000000
          3       0        22.990333
                  1        17.277778
                  2         2.166667
                  3        33.000000
                  4         8.333333
Name: Age, dtype: float64
```

In [12]:
```
data['Age'] = data['Age'].fillna(data.groupby(["Survived","Pclass","SibSp"]
data[data.isna().any(axis=1)]
```

Out[12]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
61	1	1	female	38.0	0	0	80.0	NaN
829	1	1	female	62.0	0	0	80.0	NaN

In [13]:
```
data=data.dropna().reset_index(drop=True)
```

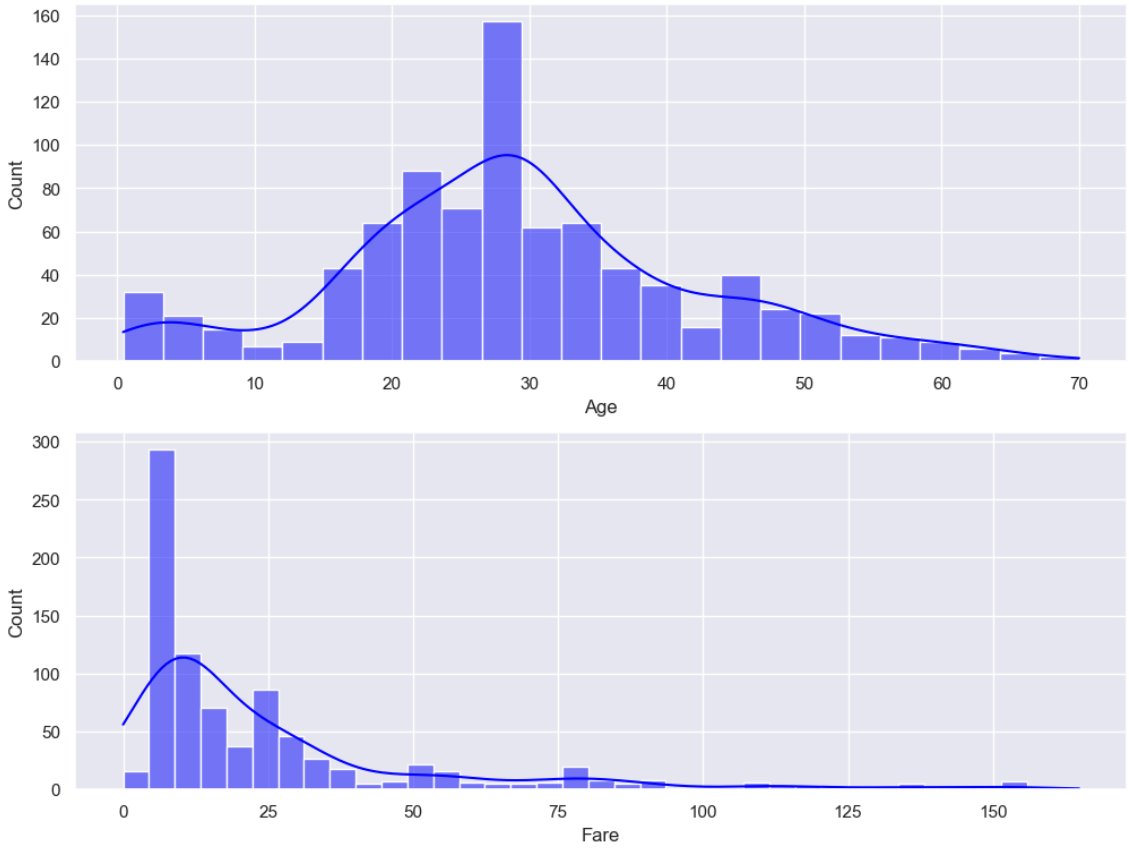
In [14]:
```
data.describe()
```

Out[14]:

	Survived	Pclass	Age	SibSp	Parch	Fare
count	882.000000	882.000000	882.000000	882.000000	882.000000	882.000000
mean	0.385488	2.306122	29.525662	0.464853	0.369615	31.799432
std	0.486986	0.835742	13.509299	0.883324	0.796919	49.781845
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	22.000000	0.000000	0.000000	7.895800
50%	0.000000	3.000000	29.103175	0.000000	0.000000	14.454200
75%	1.000000	3.000000	36.000000	1.000000	0.000000	30.500000
max	1.000000	3.000000	80.000000	5.000000	6.000000	512.329200

In [15]:
```
data = data[(np.abs(stats.zscore(data[['Age','Fare']])) < 3).all(axis=1)]
```

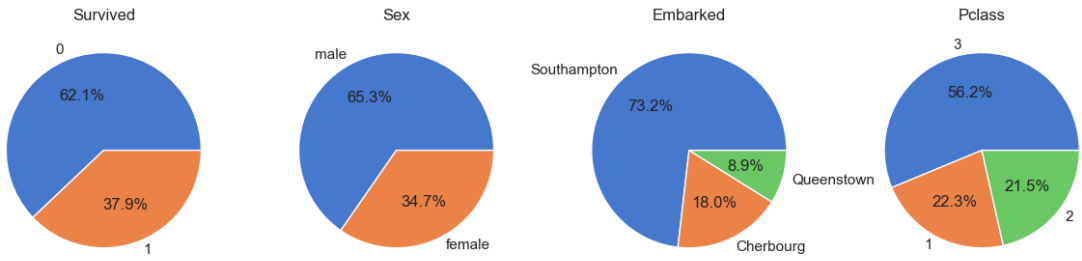
In [16]:
```
plt.figure(figsize=(12,9))
for i,col in enumerate(['Age','Fare']):
    plt.subplot(2,1,i+1)
    sns.histplot(data=data,x=col,kde=True,color="blue")
```



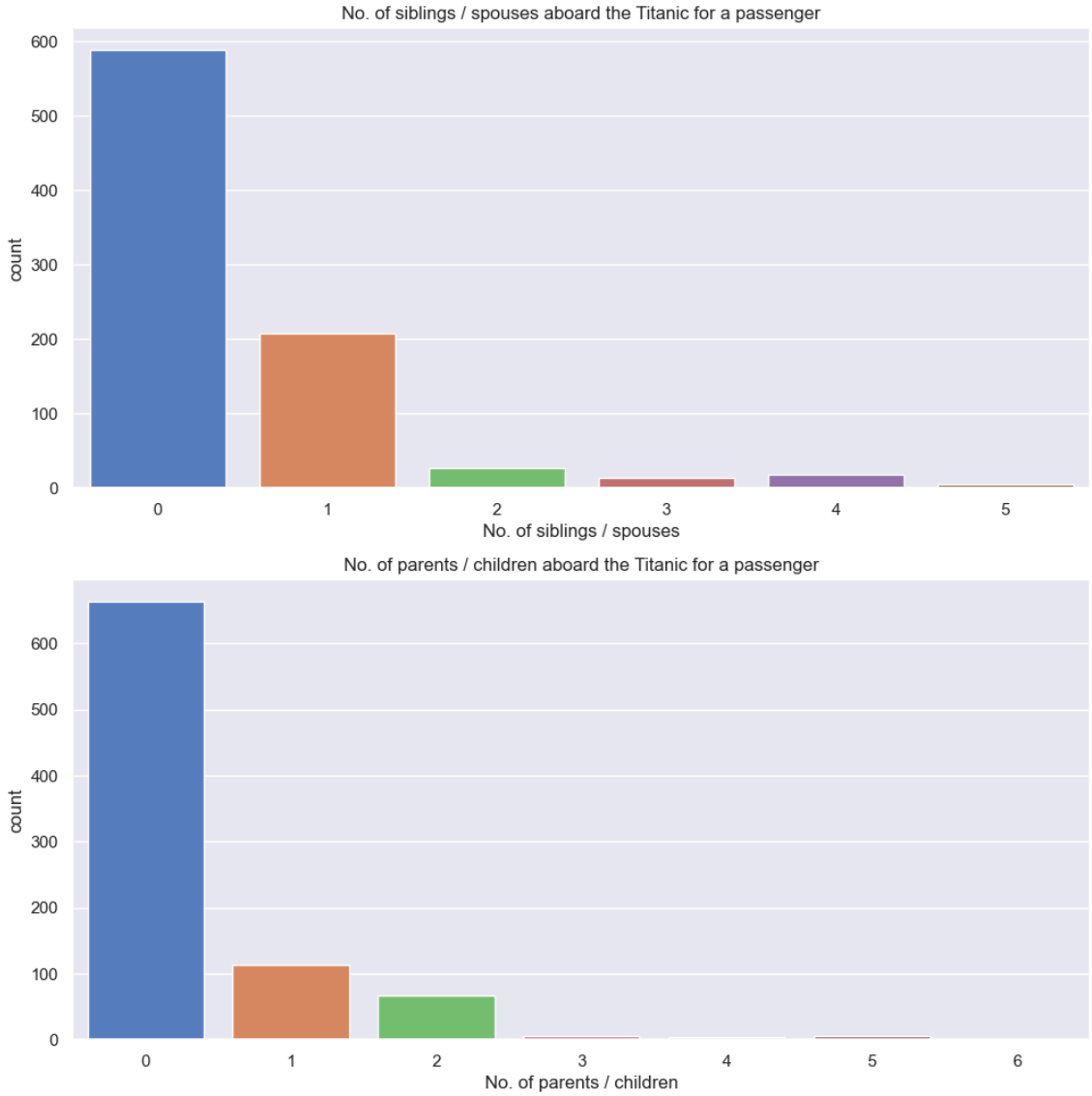
In [17]:
```
data.columns
```

Out[17]:
```
Index(['Survived', 'Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare',
       'Embarked'],
      dtype='object')
```

In [18]:
```
plt.figure(figsize=(15,12))
for i,col in enumerate(['Survived','Sex','Embarked','Pclass']):
    plt.subplot(1,4,i+1)
    x=data[col].value_counts().reset_index()
    plt.title(f"{col}")
    plt.pie(x=x['count'],labels=x[col],autopct="%0.1f%%")
```



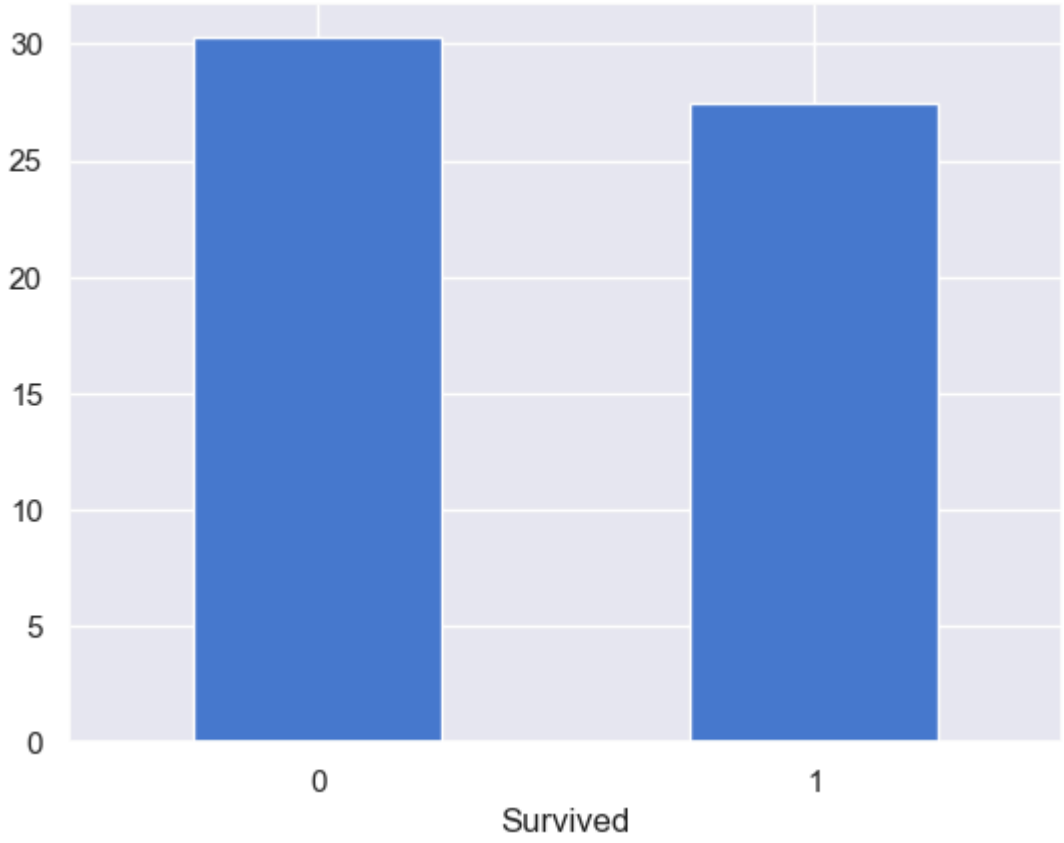
In [19]:
```
plt.figure(figsize=(12,12))

plt.subplot(2,1,1)
sns.countplot(data=data,x="SibSp")
plt.title("No. of siblings / spouses aboard the Titanic for a passenger")
plt.xlabel("No. of siblings / spouses")

plt.subplot(2,1,2)
sns.countplot(data=data,x="Parch")
plt.title("No. of parents / children aboard the Titanic for a passenger")
plt.xlabel("No. of parents / children")

plt.show()
```



In [20]:
```
data.groupby("Survived")["Age"].mean()
```

Out[20]:
```
Survived
0    30.270726
1    27.438023
Name: Age, dtype: float64
```

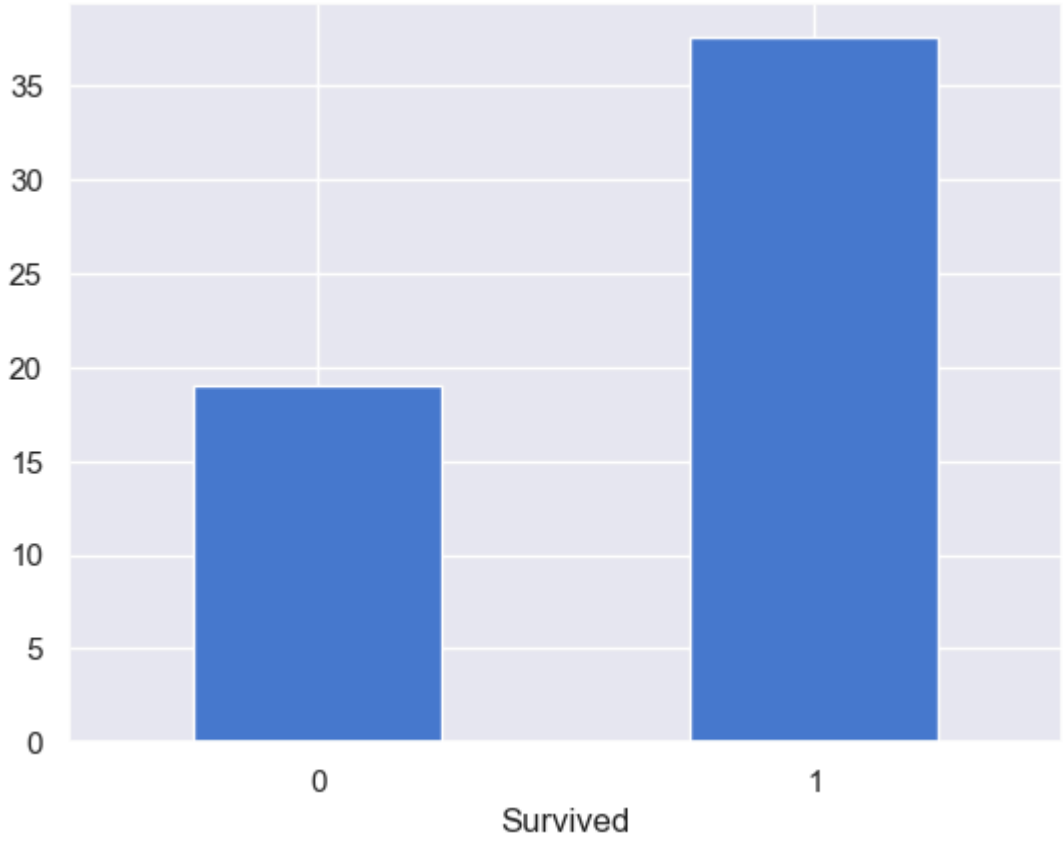
In [21]:
```
data.groupby("Survived")["Age"].mean().plot(kind="bar")
plt.xticks(rotation=0)
plt.show()
```



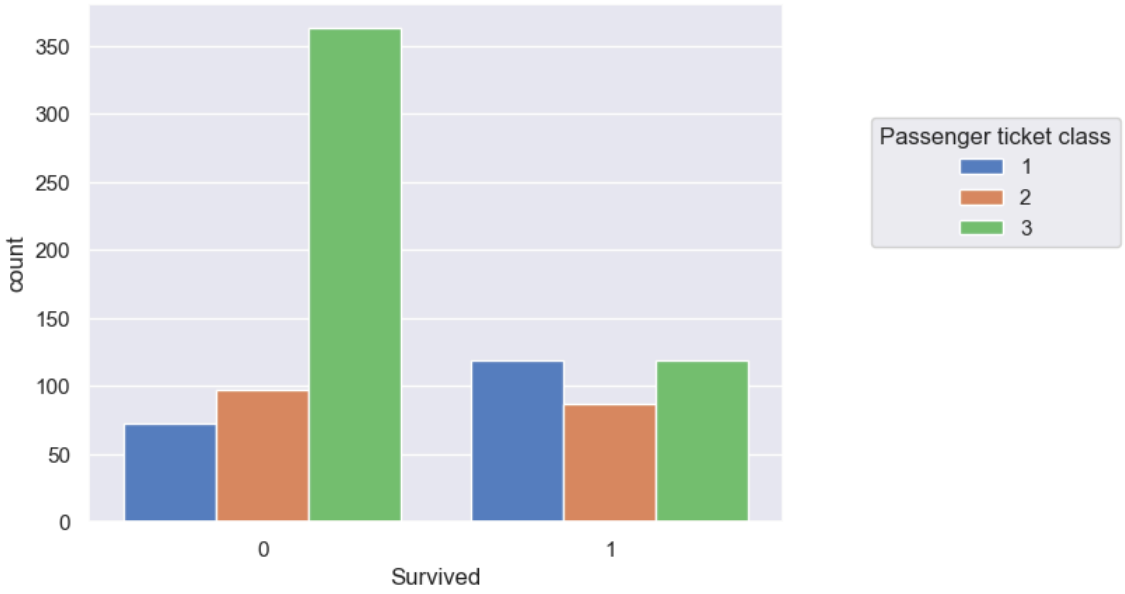
In [22]:
```
data.groupby("Survived")["Fare"].mean()
```

Out[22]:
```
Survived
0    19.026055
1    37.567334
Name: Fare, dtype: float64
```

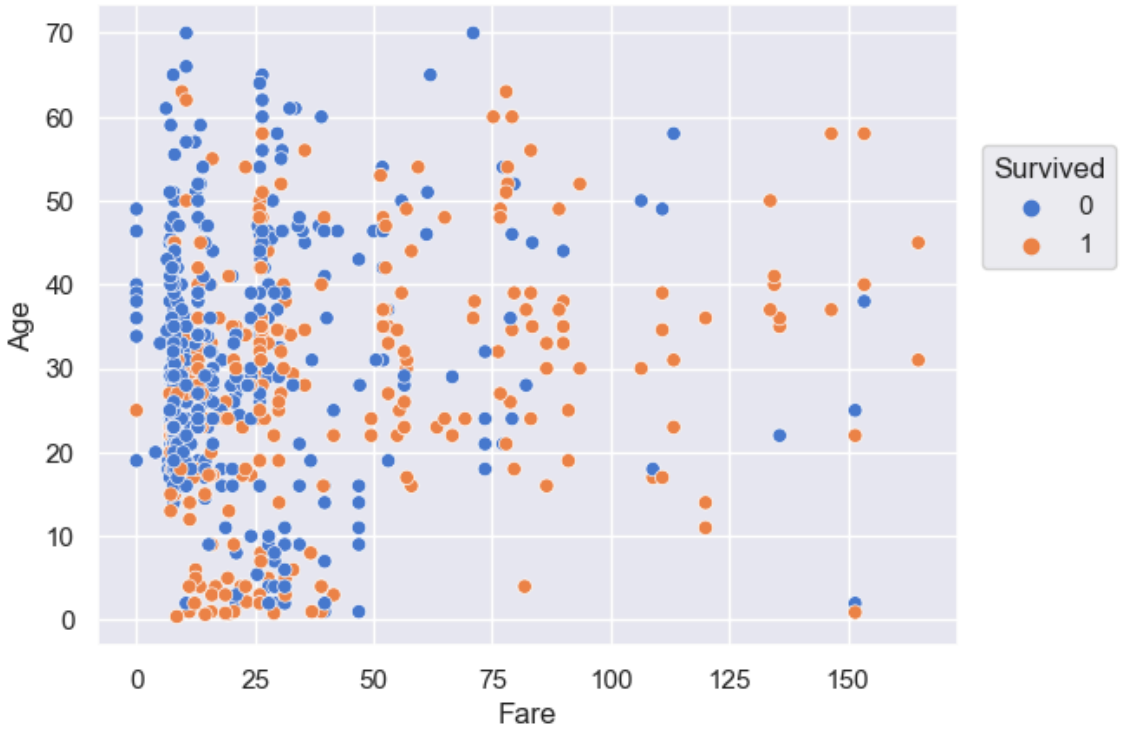
In [23]:
```
data.groupby("Survived")["Fare"].mean().plot(kind = "bar")
plt.xticks(rotation=0)
plt.show()
```



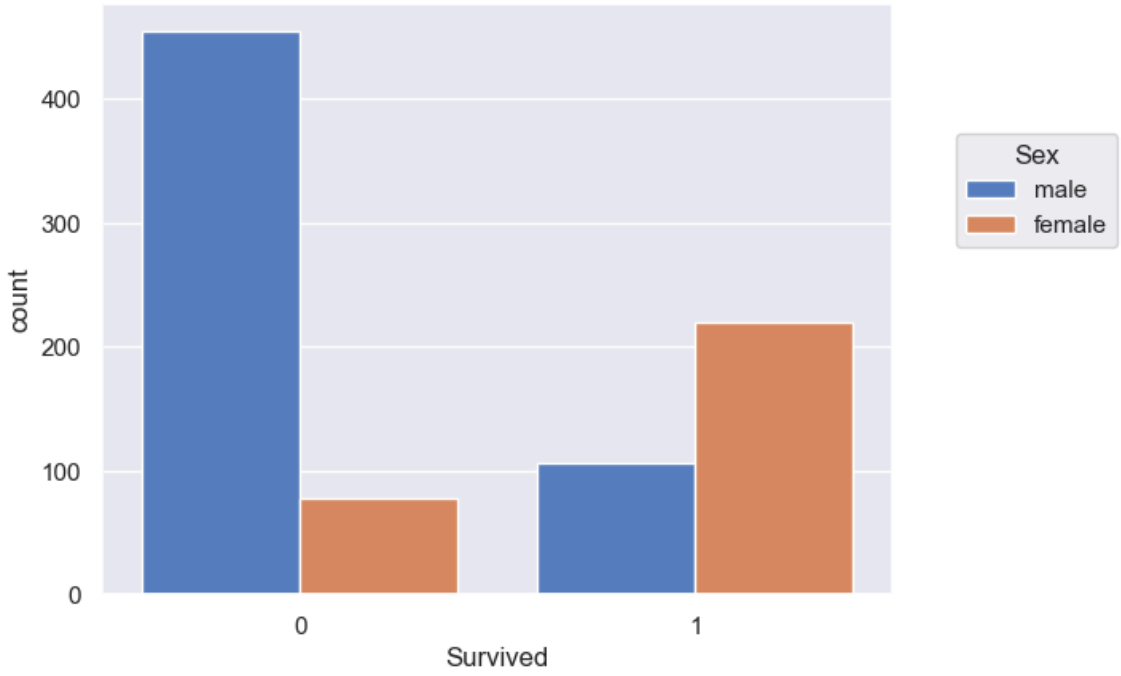
In [24]:
```
sns.countplot(data=data,x="Survived",hue="Pclass")
plt.legend(bbox_to_anchor=(1.5,0.8),title="Passenger ticket class")
plt.show()
```



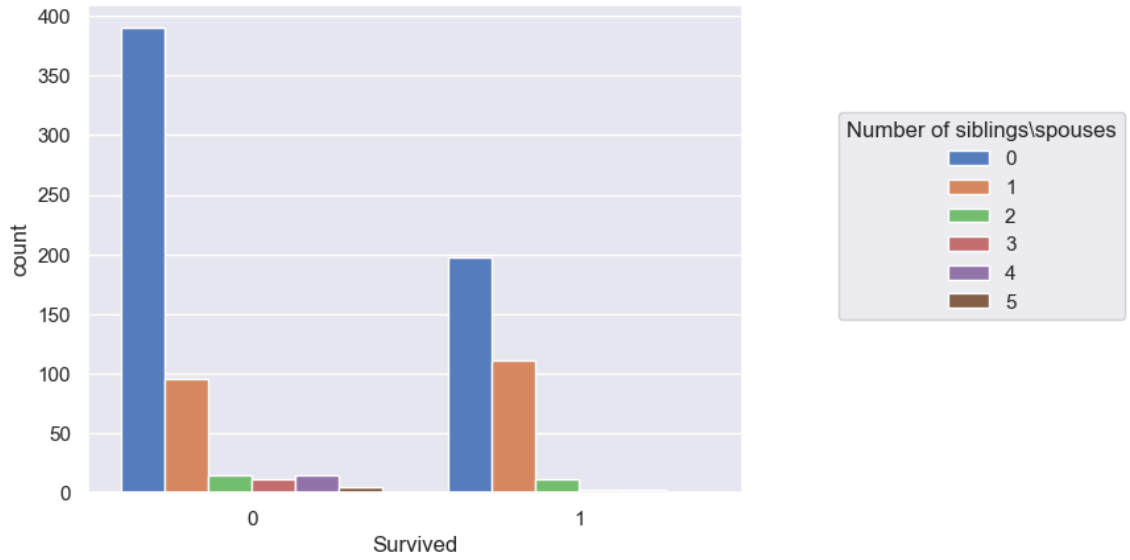
In [25]:
```
sns.scatterplot(data=data,x="Fare",y="Age",hue="Survived")
plt.legend(bbox_to_anchor=(1.2,0.8),title="Survived")
plt.show()
```



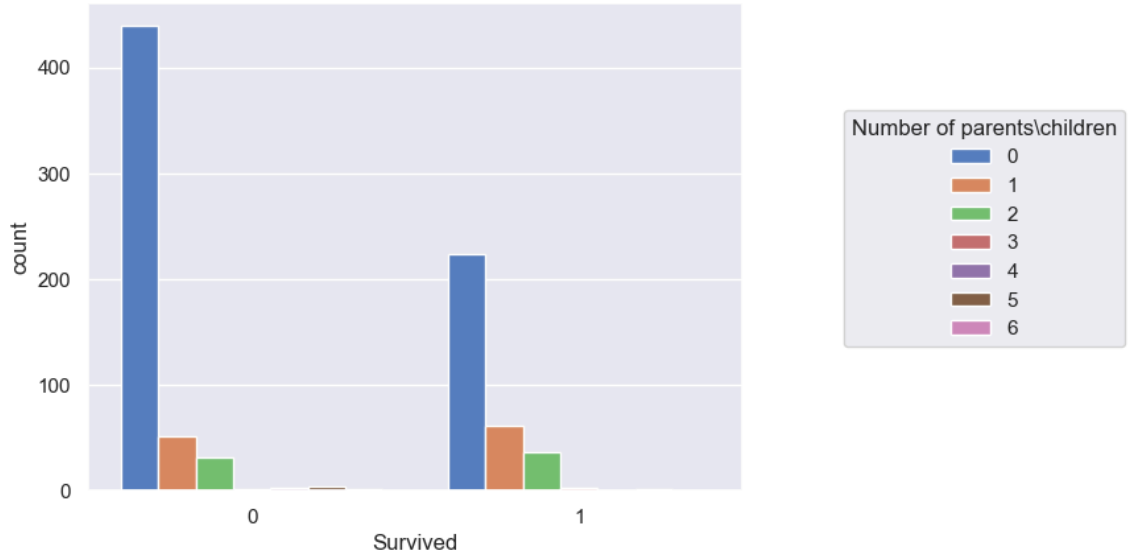
In [26]:
```
sns.countplot(data=data,x="Survived",hue="Sex")
plt.legend(bbox_to_anchor=(1.3,0.8),title="Sex")
plt.show()
```



In [27]:
```
sns.countplot(data=data,x="Survived",hue="SibSp")
plt.legend(bbox_to_anchor=(1.6,0.8),title="Number of siblings\spouses")
plt.show()
```



In [28]:
```
sns.countplot(data=data,x="Survived",hue="Parch")
plt.legend(bbox_to_anchor=(1.6,0.8),title="Number of parents\children")
plt.show()
```



In [30]:
```
pip install scikit-plot
```

```
Collecting scikit-plot
  Obtaining dependency information for scikit-plot from https://files.pyth
onhosted.org/packages/7c/47/32520e259340c140a4ad27c1b97050dd3254fdc517b1d5
9974d47037510e/scikit_plot-0.3.7-py3-none-any.whl.metadata (https://files.
pythonhosted.org/packages/7c/47/32520e259340c140a4ad27c1b97050dd3254fdc517
b1d59974d47037510e/scikit_plot-0.3.7-py3-none-any.whl.metadata)
  Downloading scikit_plot-0.3.7-py3-none-any.whl.metadata (7.1 kB)
Requirement already satisfied: matplotlib>=1.4.0 in c:\users\dhiya\anacond
a3\lib\site-packages (from scikit-plot) (3.7.2)
Requirement already satisfied: scikit-learn>=0.18 in c:\users\dhiya\anacon
da3\lib\site-packages (from scikit-plot) (1.3.0)
Requirement already satisfied: scipy>=0.9 in c:\users\dhiya\anaconda3\lib
\site-packages (from scikit-plot) (1.11.1)
Requirement already satisfied: joblib>=0.10 in c:\users\dhiya\anaconda3\li
b\site-packages (from scikit-plot) (1.2.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\dhiya\anaconda
3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\dhiya\anaconda3\li
b\site-packages (from matplotlib>=1.4.0->scikit-plot) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dhiya\anacond
a3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dhiya\anacond
a3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (1.4.4)
Requirement already satisfied: numpy>=1.20 in c:\users\dhiya\anaconda3\lib
\site-packages (from matplotlib>=1.4.0->scikit-plot) (1.24.3)
Requirement already satisfied: packaging>=20.0 in c:\users\dhiya\anaconda3
\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\dhiya\anaconda3\l
ib\site-packages (from matplotlib>=1.4.0->scikit-plot) (9.4.0)
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in c:\users\dhiya\ana
conda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\dhiya\anac
onda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (2.8.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\dhiya\anac
onda3\lib\site-packages (from scikit-learn>=0.18->scikit-plot) (2.2.0)
Requirement already satisfied: six>=1.5 in c:\users\dhiya\anaconda3\lib\si
te-packages (from python-dateutil>=2.7->matplotlib>=1.4.0->scikit-plot)
(1.16.0)
Downloading scikit_plot-0.3.7-py3-none-any.whl (33 kB)
Installing collected packages: scikit-plot
Successfully installed scikit-plot-0.3.7
Note: you may need to restart the kernel to use updated packages.
```

In [31]:
```
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import MinMaxScaler
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.model_selection import train_test_split,cross_val_score
from sklearn.metrics import classification_report
import scikitplot as skplt
```

In [32]:
```
def one_hot_encoding(data=None):
    dums = pd.get_dummies(data[["Sex","Embarked"]],dtype=int)
    dums_data = pd.concat([dums,data],axis=1).drop(columns=['Sex','Embarked
    return dums_data
```

In [33]:
```
model_data = one_hot_encoding(data)
```

In [34]:
```
model_data
```

Out[34]:

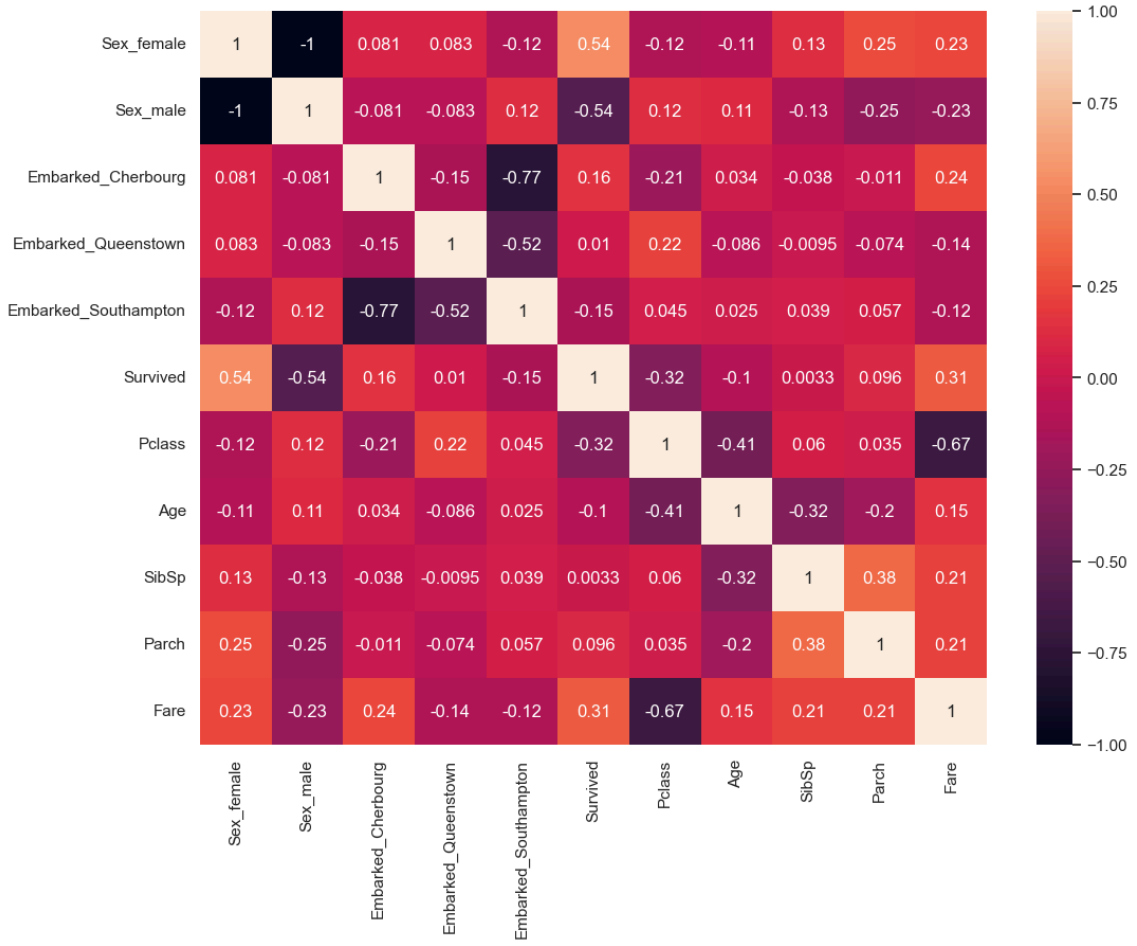
	Sex_female	Sex_male	Embarked_Cherbourg	Embarked_Queenstown	Embarked_Southa
0	0	1	0	0	
1	1	0	1	0	
2	1	0	0	0	
3	1	0	0	0	
4	0	1	0	0	
...	...	...	...	...	
852	0	1	0	0	
853	1	0	0	0	
854	1	0	0	0	
855	0	1	1	0	
856	0	1	0	1	

857 rows × 11 columns

In [35]:
```
corr = model_data.corr()
```

In [36]:
```
plt.figure(figsize=(12,9))
sns.heatmap(corr,annot=True)
```

Out[36]: <Axes: >



In [37]:
```
X = model_data.drop(columns="Survived")
y = model_data["Survived"]
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_s
```

In [38]:
```
scaler = MinMaxScaler()
scaler.fit(X[['Fare','Age']])
X[['Fare','Age']]=scaler.transform(X[['Fare','Age']])
```

In [39]:
```
gbc_model = GradientBoostingClassifier()
```

In [40]:
```
gbc_model.fit(X_train,y_train)
```

Out[40]:
```
GradientBoostingClassifier()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [41]:
```
gbc_model.score(X_train,y_train)
```

Out[41]:
```
0.9036496350364963
```

In [42]:
```
gbc_model.score(X_test,y_test)
```

Out[42]:
```
0.877906976744186
```

In [43]:
```
score = cross_val_score(gbc_model,X,y,cv=10)
avg = np.mean(score)
print(f"cross validation score for Gradient Boost:{score}")
print(f"average cross validation score for Gradient Boost:{avg}\n")
```

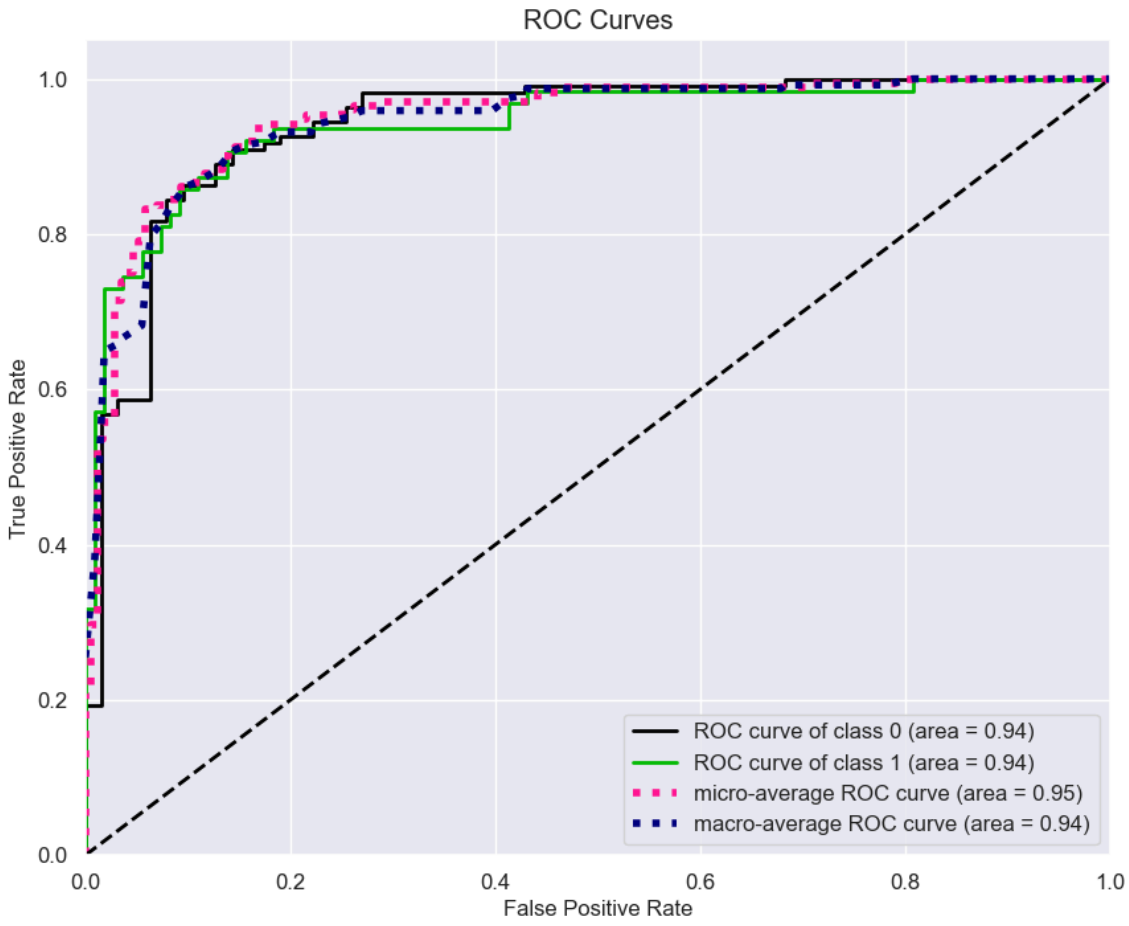
```
cross validation score for Gradient Boost:[0.8372093  0.8255814  0.8023255
8 0.88372093 0.86046512 0.87209302
 0.86046512 0.83529412 0.84705882 0.85882353]
average cross validation score for Gradient Boost:0.8483036935704513
```

In [44]:
```
y_predicted = gbc_model.predict(X_test)
y_proba=gbc_model.predict_proba(X_test)
```

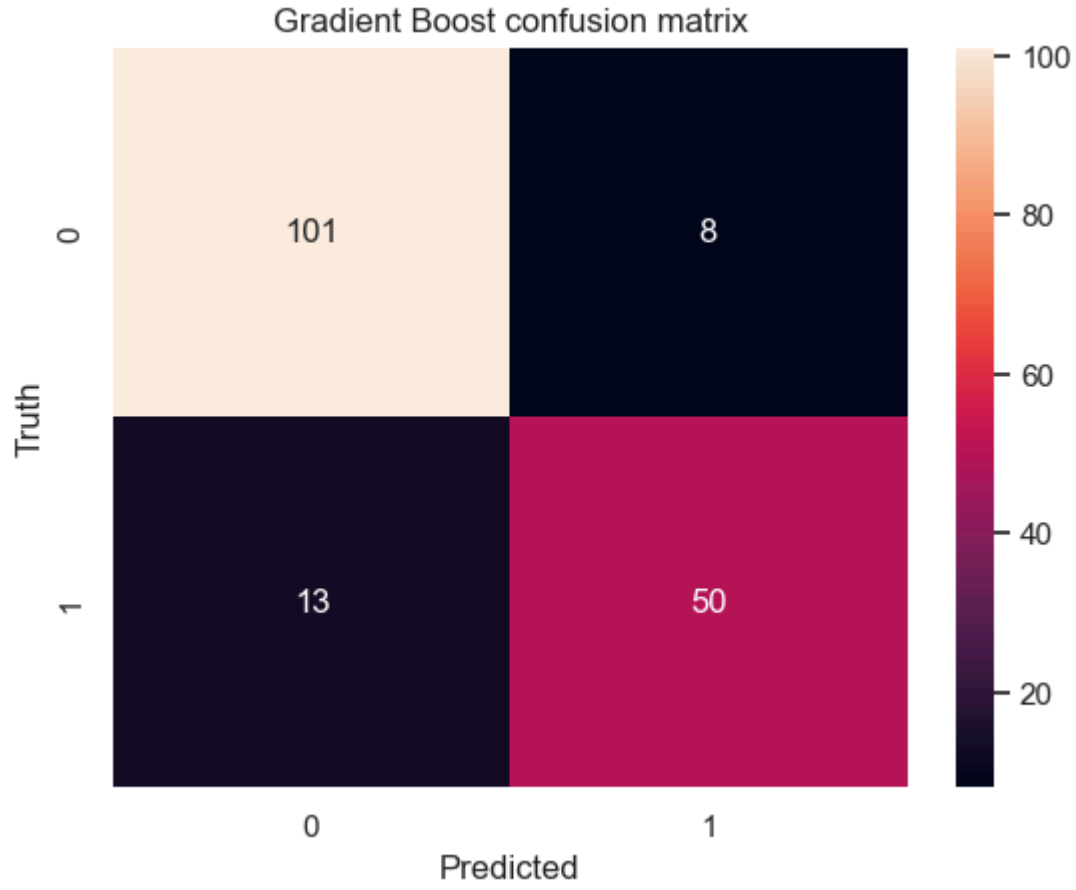
In [45]:
```
print(classification_report(y_test,y_predicted))
```

```
              precision    recall  f1-score   support

           0       0.89      0.93      0.91       109
           1       0.86      0.79      0.83        63

    accuracy                           0.88       172
   macro avg       0.87      0.86      0.87       172
weighted avg       0.88      0.88      0.88       172
```

In [46]:
```
skplt.metrics.plot_roc(y_test,y_proba,figsize=(10,8))
plt.show()
```



In [47]:
```
cm = skplt.metrics.confusion_matrix(y_test, y_predicted)
sns.heatmap(cm, annot=True,fmt='d')
plt.xlabel('Predicted')
plt.ylabel('Truth')
plt.title(f"Gradient Boost confusion matrix")
plt.show()
```



In [48]:
```
new_data = pd.DataFrame({'Pclass':[3],'Sex':['male'],'Age':[52],'SibSp':[1]
                         'Embarked':['Southampton']})
```

In [49]:
```
new_data =one_hot_encoding(new_data)
new_data =new_data.reindex(columns=X.columns,fill_value=0)
new_data[['Fare','Age']]=scaler.transform(new_data[['Fare','Age']])
```

In [50]:
```
new_data
```

Out[50]:

	Sex_female	Sex_male	Embarked_Cherbourg	Embarked_Queenstown	Embarked_Southamp
0	0	1	0	0	

In [51]:
```
prediction= gbc_model.predict(new_data)
if prediction == 0:
    print("Passenger didn't survive")
else:
    print("Passenger survived")
```

```
Passenger didn't survive
```

In [ ]: