In [120]:
```
import numpy as np
import pandas as pd
import seaborn as sbn
import matplotlib.pyplot as plt
```

In [121]:
```
df = pd.read_csv("C:/Users/olive/Desktop/Data/train.csv")
```

In [122]:
```
df.head()
```

Out[122]:

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

In [123]:
```
df.shape
```

Out[123]: (891, 12)

In [124]:

```
for i in df.groupby('Pclass'):
 print(i)
```

```
(1,      PassengerId  Survived  Pclass  \
1              2         1       1
3              4         1       1
6              7         0       1
11            12         1       1
23            24         1       1
..           ...       ...     ...
871          872         1       1
872          873         0       1
879          880         1       1
887          888         1       1
889          890         1       1

                                                  Name     Sex   Age  SibSp  \
1    Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
6                              McCarthy, Mr. Timothy J    male  54.0      0
11                            Bonnell, Miss. Elizabeth  female  58.0      0
23                        Sloper, Mr. William Thompson    male  28.0      0
..                                                 ...     ...   ...    ...
871   Beckwith, Mrs. Richard Leonard (Sallie Monypeny)  female  47.0      1
872                           Carlsson, Mr. Frans Olof    male  33.0      0
879      Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)  female  56.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
889                              Behr, Mr. Karl Howell    male  26.0      0

     Parch    Ticket     Fare        Cabin Embarked
1        0  PC 17599  71.2833          C85        C
3        0    113803  53.1000         C123        S
6        0     17463  51.8625          E46        S
11       0    113783  26.5500         C103        S
23       0    113788  35.5000           A6        S
..     ...       ...      ...          ...      ...
871      1     11751  52.5542          D35        S
872      0       695   5.0000  B51 B53 B55        S
879      1     11767  83.1583          C50        C
887      0    112053  30.0000          B42        S
889      0    111369  30.0000         C148        C

[216 rows x 12 columns])
(2,      PassengerId  Survived  Pclass  \
9             10         1       2
15            16         1       2
17            18         1       2
20            21         0       2
21            22         1       2
..           ...       ...     ...
866          867         1       2
874          875         1       2
880          881         1       2
883          884         0       2
886          887         0       2

                                             Name     Sex   Age  SibSp  Parch  \
9             Nasser, Mrs. Nicholas (Adele Achem)  female  14.0      1      0
15               Hewlett, Mrs. (Mary D Kingcome)   female  55.0      0      0
17                    Williams, Mr. Charles Eugene    male   NaN      0      0
20                            Fynney, Mr. Joseph J    male  35.0      0      0
21                           Beesley, Mr. Lawrence    male  34.0      0      0
..                                            ...     ...   ...    ...    ...
866                   Duran y More, Miss. Asuncion  female  27.0      1      0
874          Abelson, Mrs. Samuel (Hannah Wizosky)  female  28.0      1      0
880  Shelley, Mrs. William (Imanita Parrish Hall)  female  25.0      0      1
883                   Banfield, Mr. Frederick James    male  28.0      0      0
886                           Montvila, Rev. Juozas    male  27.0      0      0
```

```
                Ticket     Fare Cabin Embarked
9               237736  30.0708   NaN        C
15              248706  16.0000   NaN        S
17              244373  13.0000   NaN        S
20              239865  26.0000   NaN        S
21              248698  13.0000   D56        S
..                 ...      ...   ...      ...
866     SC/PARIS 2149  13.8583   NaN        C
874         P/PP 3381  24.0000   NaN        C
880            230433  26.0000   NaN        S
883  C.A./SOTON 34068  10.5000   NaN        S
886            211536  13.0000   NaN        S

[184 rows x 12 columns])
(3,      PassengerId  Survived  Pclass                                       Name  \
0              1         0       3                    Braund, Mr. Owen Harris
2              3         1       3                     Heikkinen, Miss. Laina
4              5         0       3                   Allen, Mr. William Henry
5              6         0       3                           Moran, Mr. James
7              8         0       3             Palsson, Master. Gosta Leonard
..           ...       ...     ...                                        ...
882          883         0       3               Dahlberg, Miss. Gerda Ulrika
884          885         0       3                    Sutehall, Mr. Henry Jr
885          886         0       3       Rice, Mrs. William (Margaret Norton)
888          889         0       3  Johnston, Miss. Catherine Helen "Carrie"
890          891         0       3                        Dooley, Mr. Patrick

        Sex   Age  SibSp  Parch            Ticket     Fare Cabin Embarked
0      male  22.0      1      0         A/5 21171   7.2500   NaN        S
2    female  26.0      0      0  STON/O2. 3101282   7.9250   NaN        S
4      male  35.0      0      0            373450   8.0500   NaN        S
5      male   NaN      0      0            330877   8.4583   NaN        Q
7      male   2.0      3      1            349909  21.0750   NaN        S
..      ...   ...    ...    ...               ...      ...   ...      ...
882  female  22.0      0      0              7552  10.5167   NaN        S
884    male  25.0      0      0   SOTON/OQ 392076   7.0500   NaN        S
885  female  39.0      0      5            382652  29.1250   NaN        Q
888  female   NaN      1      2        W./C. 6607  23.4500   NaN        S
890    male  32.0      0      0            370376   7.7500   NaN        Q

[491 rows x 12 columns])
```

In [125]:
```
df.groupby('Pclass').mean()
```

Out[125]:

Pclass	PassengerId	Survived	Age	SibSp	Parch	Fare
1	461.597222	0.629630	38.233441	0.416667	0.356481	84.154687
2	445.956522	0.472826	29.877630	0.402174	0.380435	20.662183
3	439.154786	0.242363	25.140620	0.615071	0.393075	13.675550

In [126]:
```
df.Survived.value_counts()
```

Out[126]:
```
0    549
1    342
Name: Survived, dtype: int64
```

In [127]:
```
df = df.drop(columns=['Name', 'PassengerId', 'Cabin', 'Ticket'])
```

In [128]:
```
df
```

Out[128]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	male	22.0	1	0	7.2500	S
1	1	1	female	38.0	1	0	71.2833	C
2	1	3	female	26.0	0	0	7.9250	S
3	1	1	female	35.0	1	0	53.1000	S
4	0	3	male	35.0	0	0	8.0500	S
...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S
887	1	1	female	19.0	0	0	30.0000	S
888	0	3	female	NaN	1	2	23.4500	S
889	1	1	male	26.0	0	0	30.0000	C
890	0	3	male	32.0	0	0	7.7500	Q

891 rows × 8 columns

In [129]:
```
df.columns
```

Out[129]:
```
Index(['Survived', 'Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare',
       'Embarked'],
      dtype='object')
```

In [130]:
```
df.fillna({'Age':df.Age.median(),
 'Embarked':'S'},inplace=True)
```

In [131]:
```
df
```

Out[131]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	male	22.0	1	0	7.2500	S
1	1	1	female	38.0	1	0	71.2833	C
2	1	3	female	26.0	0	0	7.9250	S
3	1	1	female	35.0	1	0	53.1000	S
4	0	3	male	35.0	0	0	8.0500	S
...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S
887	1	1	female	19.0	0	0	30.0000	S
888	0	3	female	28.0	1	2	23.4500	S
889	1	1	male	26.0	0	0	30.0000	C
890	0	3	male	32.0	0	0	7.7500	Q

891 rows × 8 columns

In [132]:
```
df.isnull().sum()
```

Out[132]:
```
Survived    0
Pclass      0
Sex         0
Age         0
SibSp       0
Parch       0
Fare        0
Embarked    0
dtype: int64
```

In [133]:
```
df.Sex[df.Sex == 'male'] = 1
df.Sex[df.Sex == 'female'] = 0
```

```
C:\Users\olive\anaconda3\lib\site-packages\ipykernel_launcher.py:1: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  """Entry point for launching an IPython kernel.
C:\Users\olive\anaconda3\lib\site-packages\ipykernel_launcher.py:2: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
```

In [134]:
```
df.Embarked[df.Embarked == 'C'] = 0
df.Embarked[df.Embarked == 'S'] = 1
df.Embarked[df.Embarked == 'Q'] = 2
```

```
C:\Users\olive\anaconda3\lib\site-packages\ipykernel_launcher.py:1: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  """Entry point for launching an IPython kernel.
C:\Users\olive\anaconda3\lib\site-packages\ipykernel_launcher.py:2: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\olive\anaconda3\lib\site-packages\ipykernel_launcher.py:3: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  This is separate from the ipykernel package so we can avoid doing imports until
```

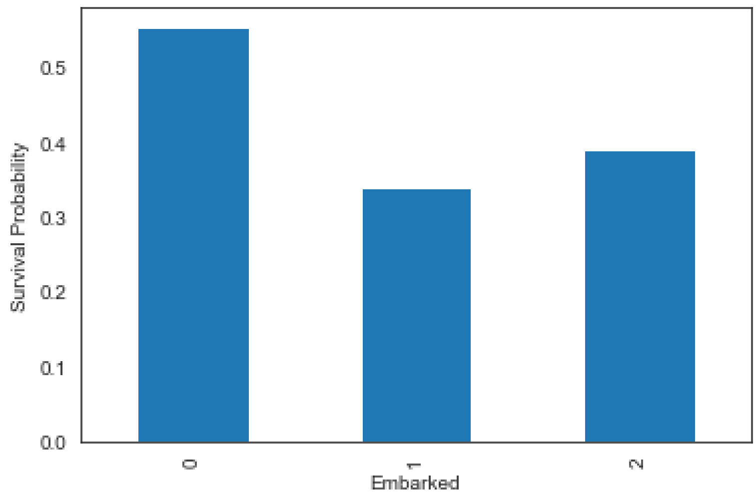
In [135]:
```
df.head()
```

Out[135]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	1	22.0	1	0	7.2500	1
1	1	1	0	38.0	1	0	71.2833	0
2	1	3	0	26.0	0	0	7.9250	1
3	1	1	0	35.0	1	0	53.1000	1
4	0	3	1	35.0	0	0	8.0500	1

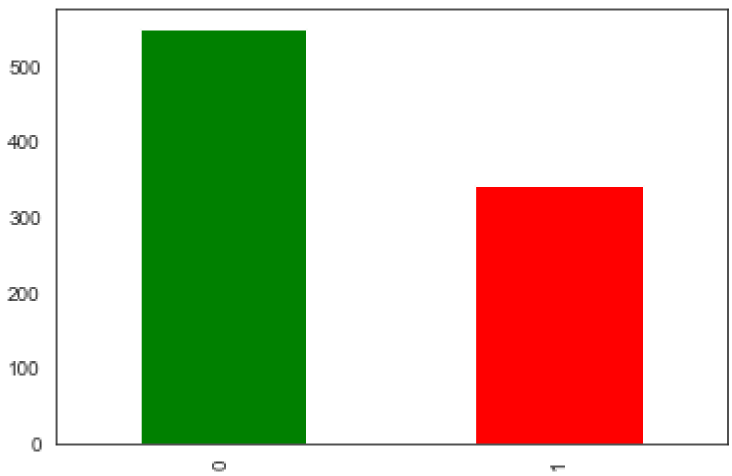
In [136]:
```
plt = df[['Embarked', 'Survived']].groupby('Embarked').mean().Survived.plot(kind="ba
r")
plt.set_xlabel('Embarked')
plt.set_ylabel('Survival Probability')
```

Out[136]: Text(0, 0.5, 'Survival Probability')



In [137]:
```
df.Survived.value_counts()
plt1 = df.Survived.value_counts().plot(kind='bar',color=('g','r'))
plt.set_xlabel('Survived-Yes or No?')
plt.set_ylabel('Passenger Count')
```

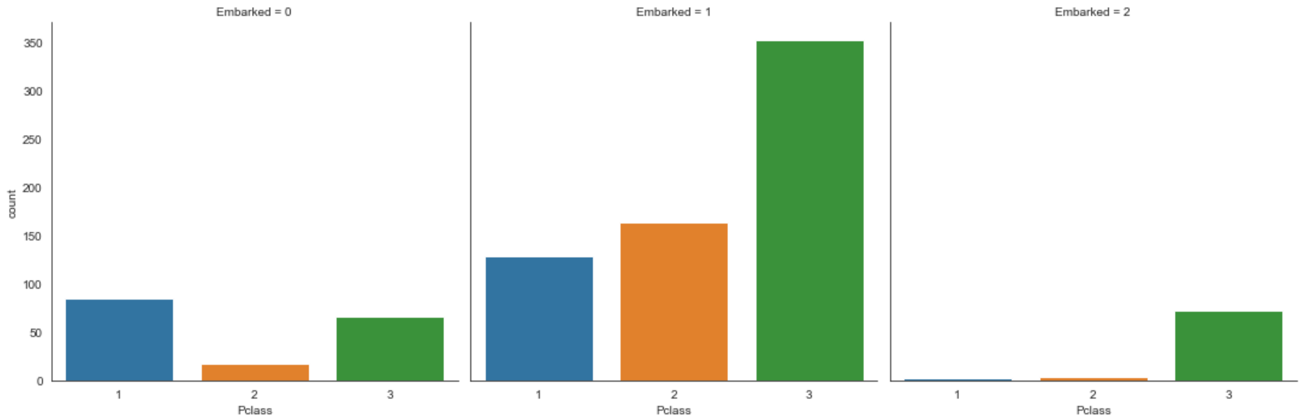
Out[137]: Text(16.200000000000003, 0.5, 'Passenger Count')



In [186]:
```
sbn.factorplot('Pclass', col = 'Embarked', data = df, kind = 'count')
```

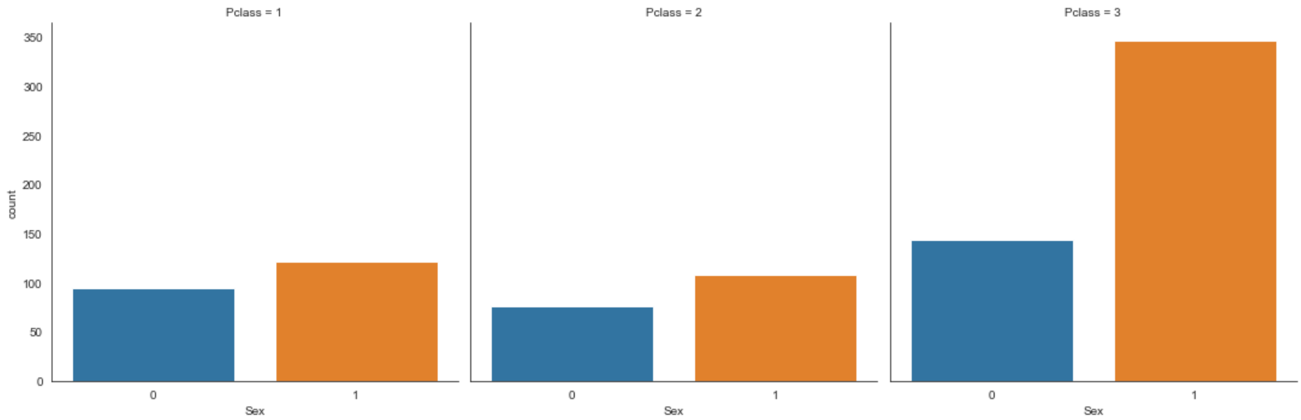
```
C:\Users\olive\anaconda3\lib\site-packages\seaborn\categorical.py:3669: UserWarning:
The `factorplot` function has been renamed to `catplot`. The original name will be r
emoved in a future release. Please update your code. Note that the default `kind` in
`factorplot` (`'point'`) has changed `'strip'` in `catplot`.
  warnings.warn(msg)
```

Out[186]: <seaborn.axisgrid.FacetGrid at 0x1e6414b2e08>



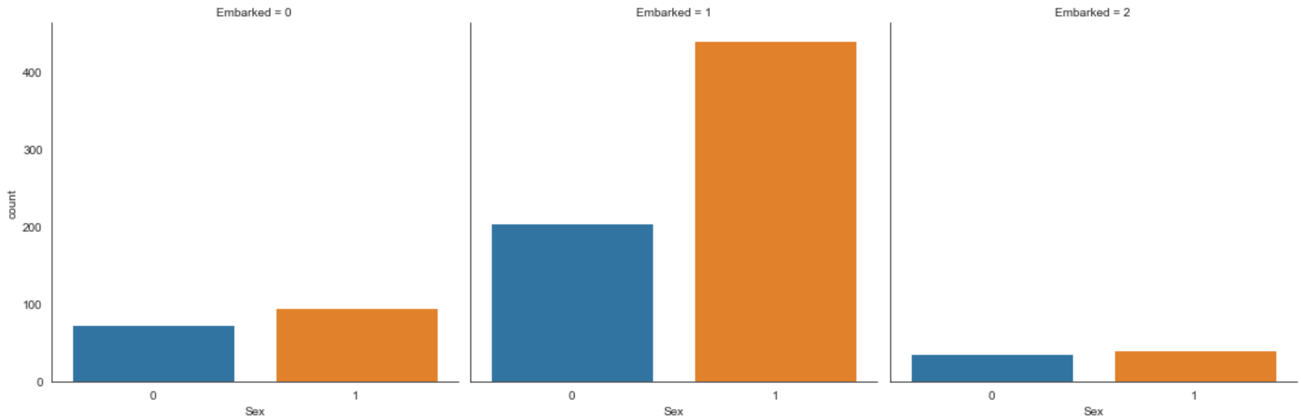
In [187]:
```
sbn.factorplot('Sex', col = 'Pclass', data = df, kind = 'count')
```

Out[187]: <seaborn.axisgrid.FacetGrid at 0x1e642fe2348>



In [188]:
```
sbn.factorplot('Sex', col = 'Embarked', data = df, kind = 'count')
```

Out[188]: <seaborn.axisgrid.FacetGrid at 0x1e64341d108>



In [138]:
```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
X1 = df.drop("Survived",axis=1)
y1 = df["Survived"]
fr = SelectKBest(chi2, k=4)
fr.fit(X1,y1)
i = fr.get_support()
X_new = pd.DataFrame(fr.transform(X1), columns = X1.columns.values[i])
```

In [139]:
```
X_new
```

Out[139]:

	Pclass	Sex	Age	Fare
0	3	1	22	7.25
1	1	0	38	71.2833
2	3	0	26	7.925
3	1	0	35	53.1
4	3	1	35	8.05
...	...	...	...	...
886	2	1	27	13
887	1	0	19	30
888	3	0	28	23.45
889	1	1	26	30
890	3	1	32	7.75

891 rows × 4 columns

In [140]:
```
X_new.columns
```

Out[140]:
```
Index(['Pclass', 'Sex', 'Age', 'Fare'], dtype='object')
```

In [141]:
```
features = X_new[['Pclass','Sex','Age','Fare']]
target = df['Survived']
```

In [142]:
```
df2 = pd.get_dummies(features[['Sex']])
```

In [143]:
```
df3 = X_new.drop(columns=['Sex'])
```

In [144]:
```
final_features = pd.concat((df2,df3),axis=1)
```

In [145]:
```
final_features
```

Out[145]:

	Sex_0	Sex_1	Pclass	Age	Fare
0	0	1	3	22	7.25
1	1	0	1	38	71.2833
2	1	0	3	26	7.925
3	1	0	1	35	53.1
4	0	1	3	35	8.05
...	...	...	...	...	...
886	0	1	2	27	13
887	1	0	1	19	30
888	1	0	3	28	23.45
889	0	1	1	26	30
890	0	1	3	32	7.75

891 rows × 5 columns

Splitting the data into Training Data and Testing Data

In [146]:
```
X = final_features.values
y = df.Survived.values
```

In [147]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(features,target,test_size=0.2,random
_state=1)
from sklearn.preprocessing import MinMaxScaler
mms = MinMaxScaler()
X_train = mms.fit_transform(X_train)
X_test = mms.transform(X_test)
```

Fitting Logistic Regression Model

In [148]:
```
from sklearn.linear_model import LogisticRegression
log = LogisticRegression(C=5,penalty='l2',class_weight=None,random_state=1)
log_ = log.fit(X_train,y_train)
log_target_prediction = log_.predict(X_test)
```

In [149]:
```
from sklearn.metrics import accuracy_score
print("Logistic Regression Score: ",accuracy_score(y_test,log_target_prediction))
a1 = accuracy_score(y_test,log_target_prediction)
```

```
Logistic Regression Score:  0.8100558659217877
```

In [150]:
```
from sklearn.metrics import mean_squared_error, r2_score
print ("MSE:",mean_squared_error(y_test,log_target_prediction))
print ("R2:",r2_score(y_test,log_target_prediction))
b1 = mean_squared_error(y_test,log_target_prediction)
c1 = r2_score(y_test,log_target_prediction)
```

```
MSE: 0.18994413407821228
R2: 0.21349185836133344
```

In [151]:
```
from sklearn.metrics import confusion_matrix
cm_dt1 = confusion_matrix(y_test, log_target_prediction)
print("The confusion matrix in case of Logistic Regression:")
print(cm_dt1)
```

```
The confusion matrix in case of Logistic Regression:
[[90 16]
 [18 55]]
```

Fitting Decision Tree Classifier

In [152]:
```
from sklearn import tree
decision_tree = tree.DecisionTreeClassifier(random_state=0,criterion = 'gini',min_sam
ples_split = 20,max_leaf_nodes=25)
decision_tree_ = decision_tree.fit(X_train,y_train)
dst_target_prediction = decision_tree_.predict(X_test)
```

In [153]:
```
print("Decision tree score: ",accuracy_score(y_test,dst_target_prediction))
a3 = accuracy_score(y_test,dst_target_prediction)
```

```
Decision tree score:  0.8212290502793296
```

In [154]:
```
print ("MSE    :",mean_squared_error(y_test,dst_target_prediction))
print ("R2     :",r2_score(y_test,dst_target_prediction))
b3 = mean_squared_error(y_test,dst_target_prediction)
c3 = r2_score(y_test,dst_target_prediction)
```

```
MSE    : 0.1787709497206704
R2     : 0.25975704316360804
```

In [155]:
```
cm_dt3 = confusion_matrix(y_test, dst_target_prediction)
print("The confusion matrix in case of Decision tree:")
print(cm_dt3)
```

```
The confusion matrix in case of Decision tree:
[[99  7]
 [25 48]]
```

In [156]:
```
from sklearn.tree import export_graphviz
export_graphviz(decision_tree,out_file="C:/Users/olive/Desktop/Data/decisiontree.dot"
)
```

Fitting Random Forest Classifier

In [157]:
```
from sklearn.ensemble import RandomForestClassifier
forest = RandomForestClassifier(max_depth=7, min_samples_split=10, n_estimators=600,
random_state=0,criterion = 'gini')
forest_ = forest.fit(X_train,y_train)
forest_target_prediction = forest_.predict(X_test)
```

In [158]:
```
print("Random forest score: ",accuracy_score(y_test,forest_target_prediction))
a2 = accuracy_score(y_test,forest_target_prediction)
```

```
Random forest score:  0.8212290502793296
```

In [159]:
```
from sklearn.metrics import mean_squared_error, r2_score
print ("MSE    :",mean_squared_error(y_test,forest_target_prediction))
print ("R2     :",r2_score(y_test,forest_target_prediction))
b2 = mean_squared_error(y_test,forest_target_prediction)
c2 = r2_score(y_test,forest_target_prediction)
```

```
MSE    : 0.1787709497206704
R2     : 0.25975704316360804
```

In [160]:
```
cm_dt2 = confusion_matrix(y_test, forest_target_prediction)
print("The confusion matrix in case of Random Forest:")
print(cm_dt2)
```

```
The confusion matrix in case of Random Forest:
[[102   4]
 [ 28  45]]
```

In [161]:
```
generalized_tree_ = generalized_tree.fit(X,y)
print("Generalized tree score: ", generalized_tree_.score(X,y))
```

```
Generalized tree score:  0.8417508417508418
```

Fitting Linear Support Vector Classifier

In [162]:
```
from sklearn.svm import LinearSVC
linear_svc = LinearSVC(penalty='l2',random_state=42,max_iter=5000)
linear_svc_ =linear_svc.fit(X_train, y_train)
linear_svc_target_prediction = linear_svc_.predict(X_test)
```

In [163]:
```
print("Linear SVC score: ",accuracy_score(y_test,linear_svc_target_prediction))
a4 = accuracy_score(y_test,linear_svc_target_prediction)
```

```
Linear SVC score:  0.776536312849162
```

In [164]:
```
from sklearn.metrics import mean_squared_error, r2_score
print ("MSE    :",mean_squared_error(y_test,linear_svc_target_prediction))
print ("R2     :",r2_score(y_test,linear_svc_target_prediction))
b4 = mean_squared_error(y_test,linear_svc_target_prediction)
c4 = r2_score(y_test,linear_svc_target_prediction)
```

```
MSE    : 0.22346368715083798
R2     : 0.07469630395450999
```

In [165]:
```
cm_dt4 = confusion_matrix(y_test, linear_svc_target_prediction)
print("The confusion matrix in case of Linear SVC:")
print(cm_dt4)
```

```
The confusion matrix in case of Linear SVC:
[[90 16]
 [24 49]]
```

Fitting Perceptron Classifier

In [166]:
```
from sklearn.linear_model import Perceptron
perceptron = Perceptron(n_iter_no_change=5000,random_state=1)
perc_ = perceptron.fit(X_train, y_train)
perceptron_target_prediction = perc_.predict(X_test)
```

```
C:\Users\olive\anaconda3\lib\site-packages\sklearn\linear_model\_stochastic_gradien
t.py:557: ConvergenceWarning: Maximum number of iteration reached before convergenc
e. Consider increasing max_iter to improve the fit.
  ConvergenceWarning)
```

In [167]:
```
print("Perceptron score: ",accuracy_score(y_test,perceptron_target_prediction))
a5 = accuracy_score(y_test,perceptron_target_prediction)
```

```
Perceptron score:  0.7932960893854749
```

In [168]:
```
from sklearn.metrics import mean_squared_error, r2_score
print ("MSE    :",mean_squared_error(y_test,perceptron_target_prediction))
print ("R2     :",r2_score(y_test,perceptron_target_prediction))
b5 = mean_squared_error(y_test,perceptron_target_prediction)
c5 = r2_score(y_test,perceptron_target_prediction)
```

```
MSE     : 0.20670391061452514
R2      : 0.14409408115792177
```

In [169]:
```
cm_dt5 = confusion_matrix(y_test, perceptron_target_prediction)
print("The confusion matrix in case of Perceptron:")
print(cm_dt5)
```

```
The confusion matrix in case of Perceptron:
[[84 22]
 [15 58]]
```

Naive Bayes Classifier or Gaussian Classifier

In [170]:
```
from sklearn.naive_bayes import GaussianNB
gaussian = GaussianNB()
gaussian_ = gaussian.fit(X_train, y_train)
gaussian_target_prediction = gaussian_.predict(X_test)
```

In [171]:
```
print("Gaussian score: ",accuracy_score(y_test,gaussian_target_prediction))
a6 = accuracy_score(y_test,gaussian_target_prediction)
```

```
Gaussian score:  0.7653631284916201
```

In [172]:
```
from sklearn.metrics import mean_squared_error, r2_score
print ("MSE    :",mean_squared_error(y_test,gaussian_target_prediction))
print ("R2     :",r2_score(y_test,gaussian_target_prediction))
b6 = mean_squared_error(y_test,gaussian_target_prediction)
c6 = r2_score(y_test,gaussian_target_prediction)
```

```
MSE    : 0.2346368715083799
R2     : 0.028431119152235507
```

In [173]:
```
cm_dt6 = confusion_matrix(y_test, gaussian_target_prediction)
print("The confusion matrix in case of Gaussian:")
print(cm_dt6)
```

```
The confusion matrix in case of Gaussian:
[[86 20]
 [22 51]]
```

Support Vector Classifier

In [174]:
```
from sklearn.svm import SVC
svc = SVC(C=0.1,kernel='rbf',random_state=0)
svc_ = svc.fit(X_train, y_train)
SVC_target_prediction = svc_.predict(X_test)
```

In [175]:
```
print("SVM score: ",accuracy_score(y_test,SVC_target_prediction))
a7 = accuracy_score(y_test,SVC_target_prediction)
```

```
SVM score:  0.776536312849162
```

In [176]:
```
from sklearn.metrics import mean_squared_error, r2_score
print ("MSE    :",mean_squared_error(y_test,SVC_target_prediction))
print ("R2     :",r2_score(y_test,SVC_target_prediction))
b7 = mean_squared_error(y_test,SVC_target_prediction)
c7 = r2_score(y_test,SVC_target_prediction)
```

```
MSE    : 0.22346368715083798
R2     : 0.07469630395450999
```

In [177]:
```
cm_dt7 = confusion_matrix(y_test, SVC_target_prediction)
print("The confusion matrix in case of SVM:")
print(cm_dt7)
```

```
The confusion matrix in case of SVM:
[[90 16]
 [24 49]]
```

K Nearest Neighbors

In [178]:
```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 3,)
knn_ = knn.fit(X_train, y_train)
knn_target_prediction = knn_.predict(X_test)
```

In [179]:
```
print("KNN score: ",accuracy_score(y_test,knn_target_prediction))
a8 = accuracy_score(y_test,knn_target_prediction)
```

```
KNN score:  0.7932960893854749
```

In [180]:
```
from sklearn.metrics import mean_squared_error, r2_score
print ("MSE    :",mean_squared_error(y_test,knn_target_prediction))
print ("R2     :",r2_score(y_test,knn_target_prediction))
b8 = mean_squared_error(y_test,knn_target_prediction)
c8 = r2_score(y_test,knn_target_prediction)
```

```
MSE     : 0.20670391061452514
R2      : 0.14409408115792177
```

In [181]:
```
cm_dt8 = confusion_matrix(y_test, knn_target_prediction)
print("The confusion matrix in case of KNN:")
print(cm_dt8)
```

```
The confusion matrix in case of KNN:
[[95 11]
 [26 47]]
```

In [182]:
```
models = pd.DataFrame({
    'Model': ['Support Vector Machines', 'KNN', 'Logistic Regression',
              'Random Forest', 'Naive Bayes', 'Perceptron',
              'Linear SVC', 'Decision Tree'],
    'Score': [a7, a8, a1, a2,
              a6, a5, a4, a3],
    'MSE': [b7,b8,b1,b2,b6,
            b5,b4,b3],
     'R2': [c7,c8,c1,c2,c6,
            c5,c4,c3]})
models.sort_values(by='Score', ascending=False)
```

Out[182]:

	Model	Score	MSE	R2
3	Random Forest	0.821229	0.178771	0.259757
7	Decision Tree	0.821229	0.178771	0.259757
2	Logistic Regression	0.810056	0.189944	0.213492
1	KNN	0.793296	0.206704	0.144094
5	Perceptron	0.793296	0.206704	0.144094
0	Support Vector Machines	0.776536	0.223464	0.074696
6	Linear SVC	0.776536	0.223464	0.074696
4	Naive Bayes	0.765363	0.234637	0.028431

In [ ]: