In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn
from sklearn import preprocessing,svm,datasets,neighbors
```

In [2]:
```python
df = pd.read_csv('C:/Users/Megha Patel/Downloads/heart.csv')
print(df)
```

```
     age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0     63    1   3       145   233    1        0      150      0      2.3
1     37    1   2       130   250    0        1      187      0      3.5
2     41    0   1       130   204    0        0      172      0      1.4
3     56    1   1       120   236    0        1      178      0      0.8
4     57    0   0       120   354    0        1      163      1      0.6
..   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
298   57    0   0       140   241    0        1      123      1      0.2
299   45    1   3       110   264    0        1      132      0      1.2
300   68    1   0       144   193    1        1      141      0      3.4
301   57    1   0       130   131    0        1      115      1      1.2
302   57    0   1       130   236    0        0      174      0      0.0

     slope  ca  thal  target
0        0   0     1       1
1        0   0     2       1
2        2   0     2       1
3        2   0     2       1
4        2   0     2       1
..     ...  ..   ...     ...
298      1   0     3       0
299      1   0     3       0
300      1   2     3       0
301      1   1     3       0
302      1   1     2       0

[303 rows x 14 columns]
```

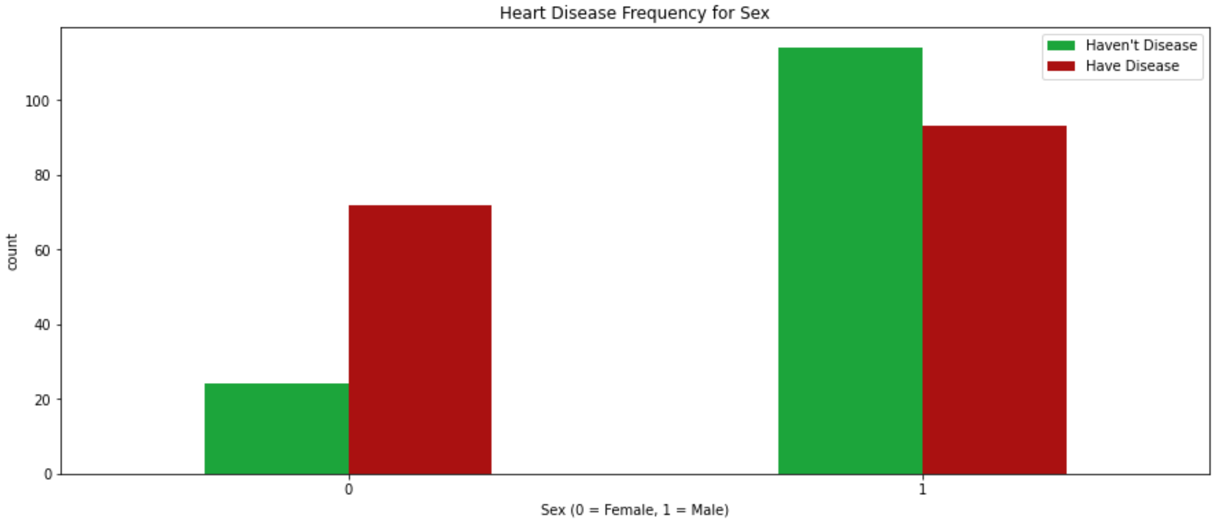
In [3]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64
 11  ca        303 non-null    int64
 12  thal      303 non-null    int64
 13  target    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

In [4]:
```python
df['sex'].value_counts()
```

Out[4]:
```
1    207
0     96
Name: sex, dtype: int64
```

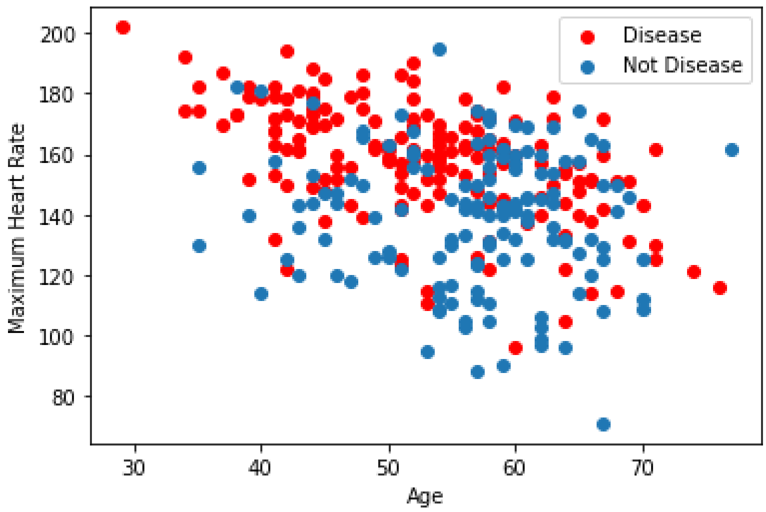
In [5]:
```
pd.crosstab(df.sex,df.target).plot(kind="bar",figsize=(15,6),color=['#1CA53B','#AA11
plt.title('Heart Disease Frequency for Sex')
plt.xlabel('Sex (0 = Female, 1 = Male)')
plt.xticks(rotation=0)
plt.legend(["Haven't Disease", "Have Disease"])
plt.ylabel('count')
plt.show()
```



In [6]:
```
df['target'].value_counts()
```

Out[6]:
```
1    165
0    138
Name: target, dtype: int64
```

In [7]:
```
plt.scatter(x=df.age[df.target==1], y=df.thalach[(df.target==1)], c="red")
plt.scatter(x=df.age[df.target==0], y=df.thalach[(df.target==0)])
plt.legend(["Disease", "Not Disease"])
plt.xlabel("Age")
plt.ylabel("Maximum Heart Rate")
plt.show()
```



In [8]:
```
# extracting the x and y from the dataset
x = df.iloc[:,:-1].values
```

```
y = df.iloc[:,-1].values
print(x)
print(y)
```

```
[[63.  1.  3. ...  0.  0.  1.]
 [37.  1.  2. ...  0.  0.  2.]
 [41.  0.  1. ...  2.  0.  2.]
 ...
 [68.  1.  0. ...  1.  2.  3.]
 [57.  1.  0. ...  1.  1.  3.]
 [57.  0.  1. ...  1.  1.  2.]]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0]
```

In [9]:

```
# feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x = sc.fit_transform(x)
print(x)
```

```
[[ 0.9521966   0.68100522  1.97312292 ... -2.27457861 -0.71442887
  -2.14887271]
 [-1.91531289  0.68100522  1.00257707 ... -2.27457861 -0.71442887
  -0.51292188]
 [-1.47415758 -1.46841752  0.03203122 ...  0.97635214 -0.71442887
  -0.51292188]
 ...
 [ 1.50364073  0.68100522 -0.93851463 ... -0.64911323  1.24459328
   1.12302895]
 [ 0.29046364  0.68100522 -0.93851463 ... -0.64911323  0.26508221
   1.12302895]
 [ 0.29046364 -1.46841752  0.03203122 ... -0.64911323  0.26508221
  -0.51292188]]
```

In [10]:

```
# splitting x and y into training and test data
from sklearn.model_selection import train_test_split
x_tr,x_te,y_tr,y_te = train_test_split(x,y,test_size = 0.2,random_state = 0)
print(x_te)
```

```
[[ 1.72421839  0.68100522 -0.93851463  0.76395577 -1.39653716 -0.41763453
   0.89896224 -1.07781984  1.43548113  1.34614673 -2.27457861 -0.71442887
   1.12302895]
 [ 1.06248543  0.68100522  1.97312292  2.19177836 -0.3722866  -0.41763453
  -1.00583187  0.23409531 -0.69663055 -0.37924438 -0.64911323 -0.71442887
   1.12302895]
 [ 0.5110413   0.68100522  1.97312292  2.19177836  0.80656782 -0.41763453
  -1.00583187  0.40901733 -0.69663055 -0.7243226  -0.64911323 -0.71442887
   1.12302895]
 [ 0.62133012  0.68100522 -0.93851463 -0.3783023   0.22680335 -0.41763453
  -1.00583187 -0.37813176  1.43548113  1.51868584 -0.64911323  0.26508221
   1.12302895]
 [ 0.84190778  0.68100522  1.00257707 -0.09273778 -0.29498467 -0.41763453
   0.89896224 -0.15947923 -0.69663055  0.65599028 -0.64911323  2.22410436
   1.12302895]
 [-0.7021358   0.68100522 -0.93851463 -0.43541521  0.53601107 -0.41763453
  -1.00583187  0.71513086 -0.69663055 -0.46551394 -0.64911323 -0.71442887
   1.12302895]
 [-1.58444641  0.68100522 -0.93851463 -1.23499586 -1.53181554 -0.41763453
  -1.00583187 -1.55885539  1.43548113  0.82852939 -0.64911323 -0.71442887
```

```
  1.12302895]
 [ 0.9521966   0.68100522 -0.93851463 -0.09273778  0.14950142 -0.41763453
  -1.00583187 -0.11574873 -0.69663055  0.31091206 -0.64911323  0.26508221
   1.12302895]
 [ 0.18017482 -1.46841752 -0.93851463  3.90516547  0.80656782  2.394438
  -1.00583187 -0.7279758   1.43548113  2.55392051 -2.27457861  1.24459328
   1.12302895]
 [ 0.9521966   0.68100522 -0.93851463 -0.09273778  1.61823808  2.394438
  -1.00583187 -0.7717063   1.43548113  0.65599028  0.97635214  2.22410436
   1.12302895]
 [ 0.29046364  0.68100522  1.00257707  1.04952029 -2.32416031  2.394438
   0.89896224  1.0212444  -0.69663055 -0.7243226   0.97635214  0.26508221
   1.12302895]
 [-0.59184697 -1.46841752 -0.93851463 -0.09273778  0.43938366 -0.41763453
   0.89896224  0.58393935 -0.69663055 -0.89686172  0.97635214 -0.71442887
  -0.51292188]
 [ 1.17277425  0.68100522 -0.93851463  0.19282673  0.14950142 -0.41763453
  -1.00583187 -0.99035883 -0.69663055  1.51868584 -0.64911323  0.26508221
   1.12302895]
 [-0.59184697  0.68100522  0.03203122 -0.09273778  0.38140721 -0.41763453
   0.89896224  0.93378339 -0.69663055 -0.37924438  0.97635214 -0.71442887
  -0.51292188]
 [-0.48155814 -1.46841752  1.00257707 -0.66386682 -0.52689046 -0.41763453
   0.89896224  0.36528682 -0.69663055  0.48345117 -0.64911323 -0.71442887
  -0.51292188]
 [-0.37126932  0.68100522  1.00257707 -1.80612489 -0.46891401 -0.41763453
   0.89896224 -0.29067075  1.43548113  0.13837295 -0.64911323 -0.71442887
  -0.51292188]
 [-0.26098049  0.68100522 -0.93851463 -1.34922166 -0.25633371  2.394438
   0.89896224 -0.11574873 -0.69663055 -0.81059216  0.97635214  2.22410436
   1.12302895]
 [-0.7021358   0.68100522  1.00257707 -0.43541521  0.1688269   2.394438
   0.89896224  1.10870541 -0.69663055 -0.89686172  0.97635214  1.24459328
  -0.51292188]
 [ 0.84190778  0.68100522 -0.93851463 -0.66386682  0.40073269 -0.41763453
   0.89896224 -2.21481297  1.43548113  0.65599028 -0.64911323  1.24459328
   1.12302895]
 [-1.36386876  0.68100522  1.00257707 -0.66386682 -0.12105533  2.394438
   0.89896224  1.939585   -0.69663055 -0.20670527 -2.27457861 -0.71442887
   1.12302895]
 [-1.36386876  0.68100522 -0.93851463  0.47839125 -0.39161208 -0.41763453
   0.89896224  1.23989692 -0.69663055 -0.89686172  0.97635214 -0.71442887
  -0.51292188]
 [-1.69473524  0.68100522 -0.93851463 -0.77809263 -0.52689046 -0.41763453
   0.89896224 -0.42186226 -0.69663055  0.13837295 -0.64911323 -0.71442887
   1.12302895]
 [ 0.84190778  0.68100522  0.03203122 -0.66386682  0.67128944 -0.41763453
  -1.00583187 -2.03989095 -0.69663055  0.31091206 -0.64911323  0.26508221
   1.12302895]
 [-0.04040284  0.68100522 -0.93851463 -1.23499586 -0.77812173 -0.41763453
  -1.00583187 -1.82123842  1.43548113 -0.89686172 -0.64911323  0.26508221
  -0.51292188]
 [-1.47415758  0.68100522  0.03203122 -1.23499586 -0.21768274 -0.41763453
   0.89896224  0.1466343  -0.69663055 -0.89686172  0.97635214 -0.71442887
  -0.51292188]
 [ 0.84190778 -1.46841752  1.00257707 -0.09273778  0.32343076 -0.41763453
   0.89896224 -2.30227398 -0.69663055  0.13837295 -0.64911323  0.26508221
   1.12302895]
 [-0.26098049  0.68100522 -0.93851463 -0.3783023  -0.66216884 -0.41763453
   0.89896224  0.80259187 -0.69663055 -0.03416616  0.97635214  1.24459328
   1.12302895]
 [-0.92271345  0.68100522 -0.93851463 -0.66386682  0.05287401 -0.41763453
  -1.00583187 -0.24694024 -0.69663055 -0.20670527  0.97635214 -0.71442887
   1.12302895]
 [-1.69473524 -1.46841752  1.00257707  0.36416545 -0.50756498 -0.41763453
   0.89896224  0.1029038  -0.69663055 -0.89686172 -0.64911323 -0.71442887
  -0.51292188]
 [-0.26098049  0.68100522  0.03203122  0.13571383 -0.87474914 -0.41763453
   0.89896224  0.36528682 -0.69663055 -0.20670527  0.97635214  0.26508221
```

```
 -0.51292188]
[ 0.62133012  0.68100522  1.00257707  0.47839125 -1.18395686 -0.41763453
 -1.00583187  0.23409531 -0.69663055  1.69122495 -0.64911323 -0.71442887
 -0.51292188]
[-1.03300228  0.68100522 -0.93851463  0.59261706  1.21240295 -0.41763453
 -1.00583187 -0.11574873  1.43548113 -0.89686172 -0.64911323  2.22410436
  1.12302895]
[-1.47415758 -1.46841752  0.03203122 -0.3211894   1.1544265  -0.41763453
  0.89896224  0.58393935 -0.69663055 -0.89686172  0.97635214 -0.71442887
 -0.51292188]
[-0.26098049  0.68100522  1.00257707  2.30600417 -0.91340011  2.394438
  0.89896224  0.54020884 -0.69663055 -0.46551394  0.97635214 -0.71442887
  1.12302895]
[-2.13589054  0.68100522  0.03203122 -0.54964101 -1.04867848 -0.41763453
  0.89896224  1.0649749  -0.69663055 -0.89686172  0.97635214 -0.71442887
 -0.51292188]
[ 0.5110413  -1.46841752 -0.93851463  2.42022998  0.05287401 -0.41763453
  0.89896224 -0.29067075  1.43548113 -0.89686172 -0.64911323 -0.71442887
 -0.51292188]
[ 1.72421839  0.68100522  1.00257707  1.62064933  0.43938366 -0.41763453
  0.89896224 -1.6463164   1.43548113  1.60495539 -0.64911323  0.26508221
  1.12302895]
[-1.03300228  0.68100522  0.03203122 -0.20696359  1.19307747 -0.41763453
 -1.00583187  0.89005288 -0.69663055 -0.89686172  0.97635214 -0.71442887
 -0.51292188]
[ 0.18017482  0.68100522 -0.93851463 -0.3783023   0.05287401  2.394438
 -1.00583187 -0.24694024  1.43548113  0.13837295 -0.64911323  0.26508221
 -0.51292188]
[ 0.84190778 -1.46841752 -0.93851463  0.47839125  0.42005817 -0.41763453
 -1.00583187  0.45274783 -0.69663055  2.20884228 -2.27457861  1.24459328
 -0.51292188]
[ 0.29046364  0.68100522 -0.93851463  0.47839125 -1.04867848 -0.41763453
  0.89896224 -0.07201822 -0.69663055 -0.55178349 -0.64911323 -0.71442887
 -2.14887271]
[ 0.29046364 -1.46841752 -0.93851463 -0.20696359  1.09645005 -0.41763453
 -1.00583187  0.40901733 -0.69663055 -0.89686172  0.97635214  0.26508221
 -0.51292188]
[-0.26098049  0.68100522  1.00257707  0.36416545 -0.44958853 -0.41763453
  0.89896224  0.84632238 -0.69663055 -0.89686172  0.97635214  3.20361543
 -0.51292188]
[-0.81242462  0.68100522 -0.93851463 -1.23499586  0.55533655 -0.41763453
 -1.00583187 -1.38393337  1.43548113 -0.03416616 -0.64911323  0.26508221
 -0.51292188]
[-0.37126932  0.68100522  1.97312292 -0.3783023  -0.64284335 -0.41763453
 -1.00583187 -1.07781984  1.43548113  0.31091206  0.97635214  0.26508221
 -0.51292188]
[ 1.72421839  0.68100522  0.03203122  1.39219771 -0.02442792 -0.41763453
 -1.00583187 -0.29067075 -0.69663055 -0.89686172  0.97635214 -0.71442887
 -0.51292188]
[-1.47415758  0.68100522  1.00257707 -0.09273778 -0.62351787 -0.41763453
 -1.00583187  0.80259187 -0.69663055  0.82852939 -0.64911323 -0.71442887
 -0.51292188]
[-0.48155814  0.68100522  1.00257707  0.47839125 -0.25633371 -0.41763453
  0.89896224  0.58393935 -0.69663055 -0.37924438 -0.64911323  0.26508221
  1.12302895]
[ 0.18017482 -1.46841752 -0.93851463  0.13571383  3.14495118 -0.41763453
 -1.00583187  0.01544279  1.43548113  0.74225984 -0.64911323  1.24459328
  1.12302895]
[-1.1432911   0.68100522  0.03203122 -0.66386682  0.32343076 -0.41763453
  0.89896224  1.0212444  -0.69663055 -0.89686172  0.97635214 -0.71442887
  1.12302895]
[-0.26098049  0.68100522  0.03203122 -0.66386682  1.52161066 -0.41763453
  0.89896224  0.97751389 -0.69663055 -0.7243226   0.97635214 -0.71442887
 -0.51292188]
[ 2.16537369 -1.46841752  0.03203122 -0.66386682  0.43938366 -0.41763453
 -1.00583187 -1.25274186  1.43548113 -0.7243226   0.97635214  0.26508221
 -0.51292188]
[-1.47415758 -1.46841752  1.00257707 -1.12077005  0.42005817 -0.41763453
 -1.00583187  0.97751389  1.43548113 -0.89686172  0.97635214 -0.71442887
```

```
 -0.51292188]
[-1.47415758  0.68100522  0.03203122  0.19282673 -0.83609818 -0.41763453
  0.89896224 -0.7717063  -0.69663055 -0.89686172 -0.64911323 -0.71442887
 -2.14887271]
[-1.69473524 -1.46841752  1.00257707 -2.14880232 -0.91340011 -0.41763453
  0.89896224  1.28362743 -0.69663055 -0.89686172  0.97635214 -0.71442887
 -0.51292188]
[-2.13589054  0.68100522 -0.93851463 -0.66386682 -0.93272559 -0.41763453
  0.89896224 -0.85916731  1.43548113  0.48345117 -0.64911323 -0.71442887
  1.12302895]
[-1.1432911  -1.46841752  1.00257707 -0.77809263 -0.08240437 -0.41763453
  0.89896224 -0.02828772 -0.69663055 -0.63805305 -0.64911323  0.26508221
 -0.51292188]
[ 0.29046364 -1.46841752  0.03203122 -0.09273778 -0.19835726 -0.41763453
 -1.00583187  1.0649749  -0.69663055 -0.89686172 -0.64911323  0.26508221
 -0.51292188]
[ 0.5110413   0.68100522  1.00257707  1.04952029 -0.66216884  2.394438
  0.89896224  0.32155632 -0.69663055  0.48345117  0.97635214 -0.71442887
 -0.51292188]
[-0.48155814 -1.46841752  0.03203122 -0.66386682 -0.0437534  -0.41763453
  0.89896224  0.54020884 -0.69663055  0.05210339  0.97635214 -0.71442887
 -0.51292188]
[ 0.40075247 -1.46841752 -0.93851463 -1.80612489  0.03354853 -0.41763453
 -1.00583187 -1.20901135 -0.69663055 -0.03416616 -0.64911323 -0.71442887
 -0.51292188]]
```

In [11]:

```
# creating and training the KNN algorithm
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=5,metric='minkowski',p=2)
classifier.fit(x_tr,y_tr)
```

Out[11]:

```
KNeighborsClassifier()
```

In [12]:

```
# predciting the output and printing
y_pred = classifier.predict(x_te)
z = np.append(arr=y_pred.reshape(61,1),values=y_te.reshape(61,1),axis=1)
print(z)
```

```
[[0 0]
 [0 1]
 [0 0]
 [0 0]
 [0 1]
 [1 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [1 1]
 [1 1]
 [0 0]
 [1 1]
 [1 1]
 [1 1]
 [0 1]
 [1 1]
 [0 0]
 [1 1]
 [1 1]
 [1 0]
 [0 0]
 [0 0]
 [1 1]
 [1 0]
 [0 0]
 [0 0]
```

```
 [1 1]
 [1 1]
 [0 0]
 [0 0]
 [1 1]
 [1 1]
 [1 1]
 [0 0]
 [0 0]
 [1 1]
 [0 0]
 [1 0]
 [1 1]
 [1 1]
 [1 1]
 [0 0]
 [1 1]
 [0 1]
 [1 1]
 [1 0]
 [0 0]
 [1 1]
 [1 1]
 [0 1]
 [1 1]
 [1 1]
 [1 1]
 [0 0]
 [1 1]
 [1 0]
 [1 1]
 [1 1]
 [1 1]]
```

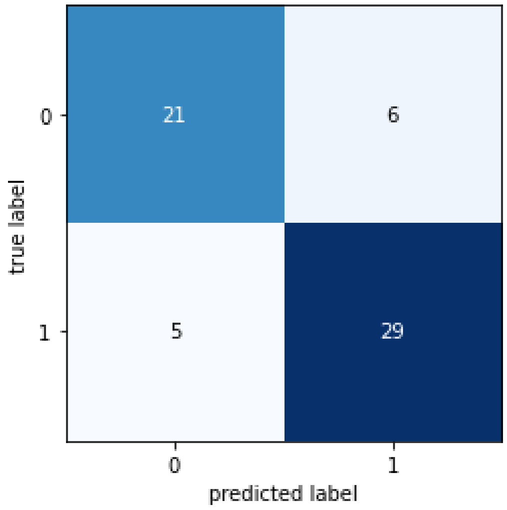
In [13]:

```
# printing the accuracy score and the confusion matrix
from sklearn.metrics import accuracy_score,confusion_matrix
acc = accuracy_score(y_te,y_pred)
cm = confusion_matrix(y_te,y_pred)
print(acc)
print(cm)
```

```
0.819672131147541
[[21  6]
 [ 5 29]]
```

In [14]:

```
# plotting the confusion matrix and the decision regions
from mlxtend.plotting import plot_confusion_matrix,plot_decision_regions
plot_confusion_matrix(cm,cmap = 'Blues')
plt.show()
```



In [16]:
```
from sklearn.svm import SVC
classifier = SVC(kernel='linear',random_state=0)
classifier.fit(x_tr,y_tr)
```

Out[16]:
```
SVC(kernel='linear', random_state=0)
```

In [17]:
```
y_pred = classifier.predict(x_te)
print(y_pred)
```

```
[0 1 1 0 0 1 0 0 0 0 1 1 0 1 1 1 0 1 0 1 1 0 0 0 1 1 0 0 1 1 1 0 1 1 1 1 0
 1 0 0 1 1 0 0 0 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1]
```

In [18]:
```
z = np.append(arr=y_pred.reshape(61,1),values=y_te.reshape(61,1),axis=1)
print(z)
```

```
[[0 0]
 [1 1]
 [1 0]
 [0 0]
 [0 1]
 [1 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [1 1]
 [1 1]
 [0 0]
 [1 1]
 [1 1]
 [1 1]
 [0 1]
 [1 1]
 [0 0]
 [1 1]
 [1 1]
 [0 0]
 [0 0]
 [0 0]
 [1 1]
 [1 0]
 [0 0]
 [0 0]
 [1 1]
 [1 1]
 [1 0]
```

```
 [0 0]
 [1 1]
 [1 1]
 [1 1]
 [1 0]
 [0 0]
 [1 1]
 [0 0]
 [0 0]
 [1 1]
 [1 1]
 [0 1]
 [0 0]
 [0 1]
 [1 1]
 [1 1]
 [1 0]
 [0 0]
 [1 1]
 [1 1]
 [1 1]
 [1 1]
 [1 1]
 [1 1]
 [0 0]
 [1 1]
 [1 0]
 [1 1]
 [1 1]
 [1 1]]
```

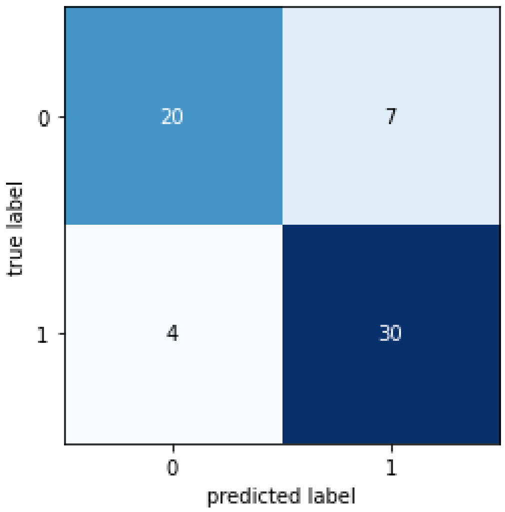
In [19]:

```
from sklearn.metrics import accuracy_score,confusion_matrix
acc= accuracy_score(y_te,y_pred)
cm = confusion_matrix(y_te,y_pred)
print(acc)
print(cm)
```

```
0.819672131147541
[[20  7]
 [ 4 30]]
```

In [20]:

```
from mlxtend.plotting import plot_confusion_matrix,plot_decision_regions
plot_confusion_matrix(cm)
plt.show()
```



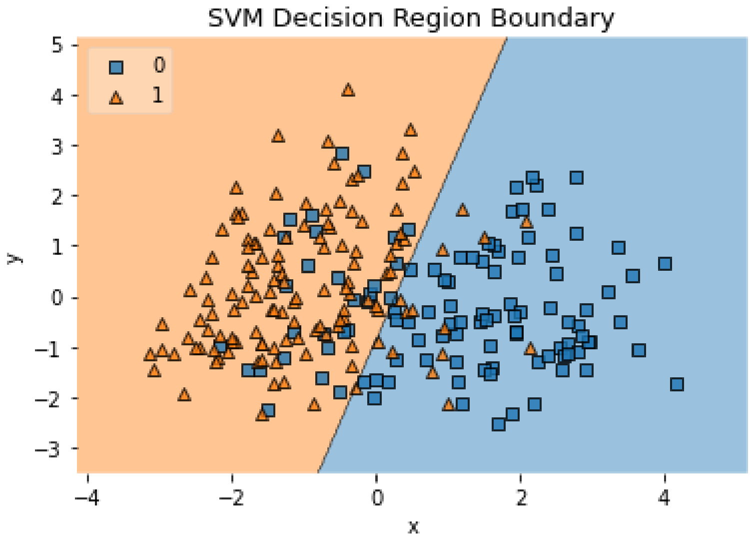
In [21]:

```
from sklearn.decomposition import PCA
from mlxtend.plotting import plot_decision_regions
```

```
clf = SVC(C=100,gamma=0.0001)
pca = PCA(n_components = 2)
X_train2 = pca.fit_transform(x_tr)
clf.fit(X_train2, y_tr)
plot_decision_regions(X_train2, y_tr, clf=clf, legend=2)

plt.xlabel('x')
plt.ylabel('y')
plt.title('SVM Decision Region Boundary', size=13)
```

Out[21]: Text(0.5, 1.0, 'SVM Decision Region Boundary')



In [22]:

```
from sklearn.decomposition import PCA
# the PCA class take n_components as the input values, since in the starting
# we do not have any information about the eigen values or the explained variance
# we do not use the n_components parameter
pca = PCA(n_components=2)
x = pca.fit_transform(x)
# to know the explained variance we have explained_variance_ration function
# of the PCA class
PVE = pca.explained_variance_ratio_
print(PVE)
# now as we know the explained valriance aof all the features we can select any
# number of features we want. for this dataset i am choosing first 2 components
print(x)
```

```
[0.21254053 0.11820708]
[[ 6.24110729e-01  2.32127028e+00]
 [-4.55987975e-01 -9.57350982e-01]
 [-1.82880491e+00  4.28847737e-02]
 [-1.71600605e+00 -4.95337323e-01]
 [-3.71356421e-01  3.01156175e-01]
 [-6.48867460e-01 -3.82882350e-01]
 [-7.26534041e-02  1.46021954e+00]
 [-1.90592574e+00 -1.15199470e+00]
 [-9.05732769e-01  1.17802505e+00]
 [-1.42452084e+00  6.00440468e-02]
 [-8.29249247e-01 -4.31111662e-01]
 [-1.76837052e+00  6.66082010e-01]
 [-1.73039364e+00 -3.62122452e-01]
 [ 4.78579542e-01 -3.66100880e-01]
 [-1.13904999e+00  3.31505033e+00]
 [-1.15305728e+00  3.38842792e-02]
 [-2.05289635e+00  1.44304375e+00]
 [ 1.24660567e+00  1.56563220e+00]
```

```
 [-1.17898795e+00 -5.93288357e-01]
 [-3.87798996e-01  2.04509682e+00]
 [-1.09484581e-02 -7.39533549e-01]
 [-1.69288377e+00 -1.05478211e+00]
 [-2.03684479e+00 -9.44816640e-01]
 [ 9.35559575e-01  9.21609753e-01]
 [-1.32664177e+00 -1.31154957e+00]
 [-1.39644919e-01  2.60255695e+00]
 [-7.38694870e-01  1.29433620e+00]
 [-1.46985273e+00 -1.28905546e+00]
 [-1.44718583e-01  4.11799172e+00]
 [ 2.33446358e-01  6.83863299e-01]
 [-2.51462707e+00 -9.04060838e-01]
 [-3.60909832e-01 -1.23607299e+00]
 [-2.29381971e+00 -4.11500959e-01]
 [ 1.95138723e-01  3.70030395e-01]
 [ 4.17978771e-02 -4.34762882e-01]
 [ 3.66186582e-01 -9.36466989e-02]
 [-1.87209672e+00  2.27149641e+00]
 [-6.12868482e-01  4.38220805e-01]
 [-8.76984260e-01  2.01456281e+00]
 [-4.29390862e-01  3.21168571e+00]
 [-5.57570271e-01  1.74775482e+00]
 [-1.26081093e+00 -1.64062709e-01]
 [ 9.11469357e-01 -2.29447547e+00]
 [-2.26867255e-01  5.82251192e-01]
 [-2.28791024e+00  6.01398620e-01]
 [-1.76066724e+00 -4.09448669e-02]
 [-2.27329407e+00  2.12736619e-01]
 [-1.70331203e+00  3.70308829e-01]
 [-1.86836572e+00  1.42913226e+00]
 [-1.28263071e+00  7.25685018e-01]
 [-1.53774483e+00  1.10788521e+00]
 [ 2.54315975e-01  3.97205643e-01]
 [ 1.04746686e+00 -7.42857672e-03]
 [-2.33560884e+00 -9.81680314e-01]
 [-1.67838555e+00  1.84323020e+00]
 [-1.15501432e+00 -5.46593725e-01]
 [-1.92948984e+00 -7.06723512e-01]
 [-2.00256825e+00 -7.52005344e-01]
 [-3.12837567e+00 -8.91051335e-01]
 [-8.06669491e-01  1.22947541e+00]
 [-3.94560324e-01  2.50933383e+00]
 [-1.27147048e+00 -7.38223238e-01]
 [-2.53287997e+00  2.47639537e-01]
 [-1.35079404e+00 -8.95169695e-01]
 [-1.36230549e+00  1.51234405e+00]
 [-2.15923721e+00 -8.48789968e-01]
 [-2.27775319e-01 -1.80679343e+00]
 [-1.34909390e+00  4.01042008e-01]
 [-2.31204282e+00 -1.13675227e+00]
 [-1.47245711e+00  1.91834447e-01]
 [-3.58290535e-01 -2.15242599e-01]
 [-9.00657970e-01 -1.98769071e+00]
 [-3.10263149e+00 -1.14139417e+00]
 [-7.77727474e-01 -4.08137174e-01]
 [-2.02255469e+00 -1.84934457e-01]
 [-3.69141309e-01  1.00307778e+00]
 [-2.17374162e-01  8.79489279e-01]
 [-7.00350164e-01 -5.38710651e-01]
 [-2.03272023e+00  2.56463178e-01]
 [ 2.97099388e-01 -1.08443288e+00]
 [-2.86242797e+00 -9.40849473e-01]
 [-1.75044772e+00  1.48892731e-01]
 [-1.73634247e+00  9.78535417e-01]
 [-5.78469356e-01  1.60771202e+00]
 [-4.45009018e-01 -6.84698902e-01]
 [ 7.36184643e-01  3.19974398e+00]
 [-2.99639697e-01  2.32348803e-01]
```

```
[-1.66919609e+00 -1.16808381e+00]
[-1.13082898e+00 -7.24242833e-02]
[ 1.94677672e-01 -1.55839503e-01]
[-1.66720309e+00  7.71701262e-01]
[-3.28562145e-01 -1.53038550e+00]
[-1.01659168e+00  3.47178140e-01]
[-1.91812262e-01  1.70969179e+00]
[-1.49262806e+00 -1.09040659e+00]
[ 9.62440887e-01 -1.09883802e+00]
[ 5.19560081e-01  2.16304236e+00]
[-3.64556302e-02 -5.17554030e-01]
[-1.19147000e+00  5.76595678e-03]
[-9.03942011e-01  1.58856809e+00]
[-1.64330093e+00  8.16829513e-01]
[ 2.21433249e+00  1.57653769e+00]
[-1.32410540e+00  1.06586014e+00]
[-9.45098926e-01 -3.44255927e-01]
[-2.20305006e+00 -5.47149621e-01]
[ 4.42831822e-01  1.06261808e+00]
[ 5.55050959e-01  2.84941902e+00]
[ 5.03643963e-02 -2.94463278e-01]
[-1.70100152e+00  3.62769746e-02]
[-1.61712759e+00  2.80553463e-02]
[ 3.82396840e-01  1.90838903e+00]
[-1.17044214e+00  4.80380395e-01]
[-5.50181431e-01  1.55192727e+00]
[-1.67849081e+00 -2.14273598e+00]
[-1.45734244e+00 -1.48979053e-01]
[-3.02772373e+00 -3.84985508e-01]
[-9.70018179e-01 -5.29495703e-01]
[-4.33119094e-01 -3.11568230e-01]
[-2.66826953e+00 -7.16422029e-01]
[ 2.84915902e-02 -1.91342721e-01]
[ 1.49653150e+00  9.69770490e-01]
[-1.19623625e+00  5.17978486e-01]
[-1.87579331e+00 -1.66117333e-01]
[-2.14878334e+00  8.53020352e-01]
[-3.43174362e+00 -1.02813770e+00]
[-3.04480004e+00 -8.49784825e-01]
[-1.53392034e+00 -1.70025940e+00]
[-1.28012604e+00  2.29544296e+00]
[-1.46794139e+00  8.87683868e-01]
[ 8.89542149e-01  1.14033501e+00]
[-1.65505163e+00  1.34730906e+00]
[-1.30826680e+00  5.03005648e-01]
[-2.06551481e+00 -7.25253420e-01]
[-2.17738669e+00 -1.47362290e+00]
[-2.22565349e+00  2.73037781e-01]
[-1.72382142e+00  1.82338061e-01]
[-7.59739493e-01  1.02525927e+00]
[-6.34687983e-01  9.96422582e-01]
[ 6.25607126e-01 -1.75577351e+00]
[ 2.10385361e+00 -1.13242627e+00]
[-1.66382947e+00  1.15509250e+00]
[-1.09304542e+00 -1.10576638e+00]
[-2.30540299e+00 -2.71697450e-01]
[-4.30092160e-01  1.63349949e-01]
[ 2.48771048e-01  1.00283690e+00]
[-2.41139543e-01  1.43371763e+00]
[-1.36885327e+00 -2.73621086e-03]
[-1.81778359e+00  1.82350637e+00]
[-2.54673833e+00 -8.09381006e-01]
[-2.26928559e+00 -1.03745572e+00]
[ 4.50718788e-01  1.15403871e+00]
[ 4.65929054e-01 -4.08170719e-01]
[ 1.03307941e-01  1.57236212e+00]
[-1.50660602e-01  2.38152957e+00]
[-1.81284743e+00  5.03633148e-02]
[-1.57907341e-01 -1.36373081e-01]
```

```
[-2.45862893e+00 -2.00484134e-01]
[-2.73927954e+00 -1.67346422e+00]
[ 8.95162473e-01 -5.29213408e-01]
[-1.05488497e+00 -2.73396006e-01]
[-5.98013045e-01 -6.19006677e-01]
[-1.21604215e+00  1.26002507e+00]
[-2.77747071e+00 -1.67406427e+00]
[-1.68346856e+00 -6.04442562e-01]
[-1.68346856e+00 -6.04442562e-01]
[ 3.42750627e+00  8.49963994e-01]
[ 2.98933554e+00 -9.51270294e-01]
[ 2.20059160e+00  1.28327713e+00]
[ 1.21227746e+00 -1.19898712e-01]
[ 2.65181760e+00 -5.65430839e-01]
[ 4.93560075e-01  1.16507761e+00]
[-2.61776747e-01 -1.68196011e+00]
[ 1.26485771e-02  1.68105218e-01]
[ 2.87370588e-01  2.50122842e-01]
[ 2.67258979e+00 -1.15735347e+00]
[ 1.32869214e+00 -3.05618567e+00]
[ 1.06495037e+00 -6.22649680e-01]
[-1.19599518e+00  1.31739406e+00]
[ 1.63210499e+00 -2.60320608e+00]
[ 1.65603152e+00  2.79448715e-01]
[ 1.90839438e+00 -7.54597839e-01]
[ 2.22220607e+00  1.16804208e+00]
[-1.03560367e+00  1.59073877e+00]
[ 3.32834850e-01 -3.74670222e-01]
[ 1.45974253e+00 -4.90202634e-01]
[-1.15018365e+00 -6.77304925e-01]
[ 1.07035164e+00 -1.17464142e+00]
[ 2.61659353e+00 -1.31513780e+00]
[-5.38173676e-01 -3.53048268e-01]
[-1.53832531e+00 -2.09653759e+00]
[ 9.94845520e-01 -5.68084873e-01]
[ 2.81843101e+00 -1.12872025e+00]
[ 1.02161781e+00 -1.77475747e+00]
[ 2.96521759e+00 -2.30563001e-01]
[ 3.26448398e-01  2.78204770e-01]
[ 3.62859705e+00  3.63324103e-01]
[ 1.70317729e-01 -2.49629513e-01]
[ 8.03973390e-01  6.06859830e-01]
[ 2.91576516e+00 -1.39438828e+00]
[-5.08161331e-01  3.28011328e-01]
[-1.81927484e+00 -1.30075367e+00]
[ 2.45124416e+00 -9.95774976e-01]
[ 1.60916279e+00 -4.00770853e-01]
[ 2.15052545e+00  2.31391554e+00]
[ 4.02011140e+00  6.46738596e-01]
[-3.98057583e-02 -1.44794737e+00]
[ 1.61848286e+00 -1.47427040e+00]
[ 1.65217809e+00  1.15863210e+00]
[ 5.94297033e-01 -1.17577198e+00]
[ 1.33672683e-01 -1.38013666e+00]
[-2.30574052e-02  1.74943974e-02]
[ 2.50244761e+00 -1.52222713e+00]
[-2.80704014e-01 -2.26958995e+00]
[ 1.60151225e+00  7.12705025e-01]
[ 1.50220402e+00  6.50097304e-02]
[ 2.03158524e+00  5.89018929e-01]
[ 1.06473628e+00  7.05765553e-01]
[ 2.69804656e+00  9.41076514e-01]
[ 2.20520701e+00 -3.64042767e-02]
[ 7.63248551e-01 -2.01888579e-01]
[ 2.92325108e+00  2.36151943e+00]
[ 4.00895722e+00 -1.95314675e+00]
[-1.83351385e-02  2.54563652e+00]
[ 4.64543392e+00  2.55026802e+00]
[ 2.07257421e+00 -2.24559036e+00]
```

```
[ 3.14268207e+00 -1.27610247e+00]
[ 1.66870959e+00 -8.60889850e-02]
[ 6.51264950e-01 -3.15972941e+00]
[-1.49252085e-01  1.80975707e+00]
[ 1.48166051e+00 -3.82242820e-01]
[-2.22063581e+00 -9.13453066e-01]
[ 2.46791133e+00  1.75059368e+00]
[ 1.91904872e+00 -2.16699666e-02]
[ 3.36660475e+00 -9.02727928e-01]
[ 2.83534111e+00  1.01099038e+00]
[ 2.44404441e-01 -1.00595328e+00]
[-2.02766445e-01  1.70739682e-01]
[ 1.07816964e+00  4.50275681e-01]
[ 1.30334294e+00  8.78937102e-01]
[-5.21969910e-01 -1.75780455e+00]
[ 2.97843850e+00  4.58581276e-01]
[ 7.91370382e-01  8.82797166e-01]
[ 1.34086441e+00  6.75245670e-01]
[ 2.80998835e+00 -8.70841338e-01]
[ 1.89414306e+00 -9.55359879e-01]
[-2.64717957e-01 -7.88012595e-01]
[ 2.33236000e+00  1.08946575e+00]
[ 1.66749048e+00  8.28025663e-01]
[-6.47467855e-01  1.82658564e+00]
[ 1.76295131e+00  1.13012802e+00]
[ 3.53258245e+00 -1.10586565e+00]
[ 1.90690702e+00 -3.61745363e-01]
[ 2.22080478e+00  1.95187504e+00]
[ 2.03223384e+00 -5.67301300e-01]
[-7.21984983e-01  1.75405581e+00]
[ 1.63809100e+00 -5.51876851e-01]
[ 2.95613650e+00 -9.57498551e-01]
[ 1.49043380e+00 -1.43444058e+00]
[ 9.38365196e-01  3.55678315e-01]
[-1.20144058e-01 -1.84943873e+00]
[ 2.09527012e+00  1.92708014e+00]
[-1.37985027e+00 -1.13851558e+00]
[ 2.81468433e+00 -1.63942506e+00]
[ 9.95478319e-01 -2.57026849e-01]
[ 1.19198297e+00 -1.66402855e+00]
[ 6.24205035e-01 -8.78329825e-01]
[ 2.35384370e+00  7.40691621e-01]
[-4.50054471e-01 -6.62998751e-01]
[ 2.77991960e+00 -8.24956072e-01]
[ 3.23628081e+00 -2.27227446e-01]
[-5.40748692e-01 -1.16363059e+00]
[ 4.72063927e-01  5.54206652e-01]
[ 1.57253846e+00 -7.36114933e-01]
[-8.16632677e-01 -1.48838837e+00]
[ 1.28581045e+00 -1.50254445e+00]
[-4.36931251e-01 -1.19396153e+00]
[ 1.86431121e+00 -7.87864065e-01]
[-7.26512444e-01 -6.06163516e-01]
[-3.15969043e-01  2.83672363e+00]
[ 2.63158109e+00 -1.03591185e+00]
[ 7.64330737e-01 -9.90866377e-01]
[-2.35336603e-01 -1.62417337e-01]
[ 1.80690338e-01  9.72556211e-01]
[-1.67832632e+00 -1.07337582e+00]
[ 1.63929145e+00 -8.25868484e-01]
[ 2.28872040e+00 -1.22909370e+00]
[-1.24386827e+00  4.70956394e-01]
[-8.56949860e-01  8.73505515e-01]
[ 2.15706459e+00 -1.38606749e+00]
[ 1.41816441e+00 -1.54885040e+00]
[-4.06059853e-01 -3.49594568e-01]
[ 2.16927742e+00 -5.31447414e-01]
[ 2.24175970e+00  2.27901051e+00]
[ 4.14157859e-01  8.36052633e-01]
```

```
[ 9.67660777e-01 -2.38918560e+00]
[ 2.58185522e+00 -8.59185999e-01]
[ 4.14314275e-01 -5.79482296e-01]
[ 1.89555409e+00  1.46786667e+00]
[ 1.14667187e+00 -5.19529495e-01]
[-7.08592873e-01 -1.04575189e+00]
[ 2.45900545e+00  4.78261911e-01]
[ 1.76275536e+00 -2.33681621e+00]
[-8.60056772e-01  1.06851556e+00]]
```

In [23]:

```
# splitting the dataset into training and test set for the svm algorithm
from sklearn.model_selection import train_test_split
x_tr,x_te,y_tr,y_te = train_test_split(x,y,test_size= 0.2,random_state=0)
```

In [25]:

```
from sklearn.svm import SVC
classifier = SVC(kernel='linear',random_state=0)
classifier.fit(x_tr,y_tr)
```

Out[25]:

```
SVC(kernel='linear', random_state=0)
```

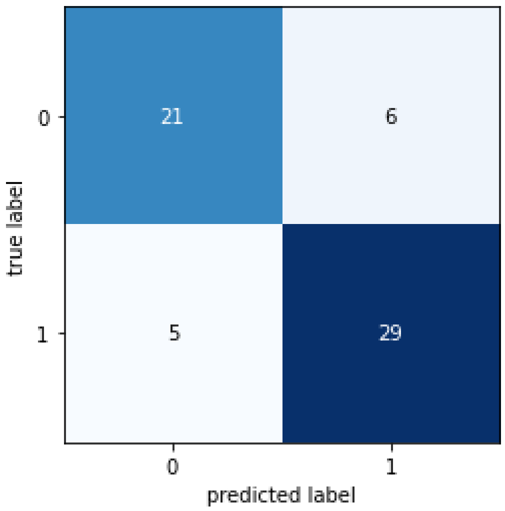
In [26]:

```
y_pred = classifier.predict(x_te)
from sklearn.metrics import accuracy_score,confusion_matrix
acc = accuracy_score(y_te,y_pred)
cm = confusion_matrix(y_te,y_pred)
print(acc)
print(cm)
```

```
0.819672131147541
[[21  6]
 [ 5 29]]
```

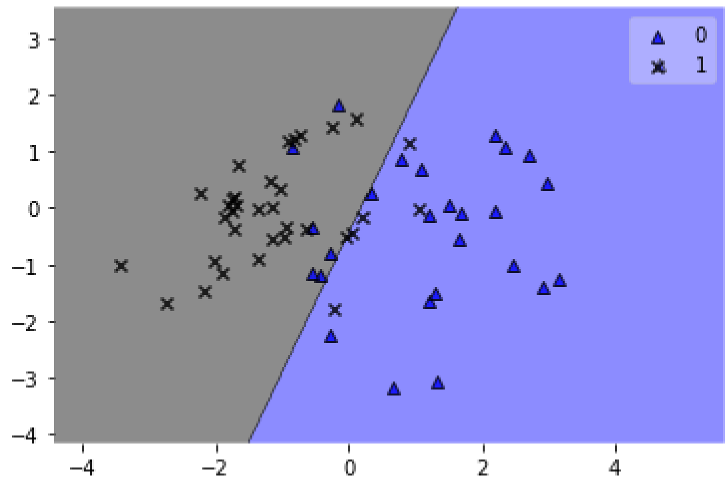
In [27]:

```
from mlxtend.plotting import plot_confusion_matrix,plot_decision_regions
plot_confusion_matrix(cm)
plt.show()
```



In [28]:

```
plot_decision_regions(X=x_te,y=y_te,clf=classifier,colors='blue,black',markers='^x')
plt.show()
```



In [29]:

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=5,metric='minkowski',p=2)
classifier.fit(x_tr,y_tr)
```

Out[29]: KNeighborsClassifier()

In [30]:

```
# predciting the output and printing
y_pred = classifier.predict(x_te)
z = np.append(arr=y_pred.reshape(61,1),values=y_te.reshape(61,1),axis=1)
print(z)
```

```
[[0 0]
 [1 1]
 [1 0]
 [0 0]
 [0 1]
 [1 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [1 1]
 [1 1]
 [0 0]
 [1 1]
 [1 1]
 [0 1]
 [1 1]
 [1 1]
 [0 0]
 [1 1]
 [1 1]
 [0 0]
 [0 0]
 [0 0]
 [1 1]
 [0 0]
 [1 0]
 [1 0]
 [1 1]
 [1 1]
 [0 0]
 [0 0]
 [1 1]
 [1 1]
 [1 1]
 [0 0]
 [0 0]
```

```
 [1 1]
 [0 0]
 [0 0]
 [1 1]
 [1 1]
 [1 1]
 [0 0]
 [1 1]
 [1 1]
 [1 1]
 [1 0]
 [0 0]
 [1 1]
 [1 1]
 [1 1]
 [1 1]
 [1 1]
 [1 1]
 [0 0]
 [1 1]
 [1 0]
 [1 1]
 [1 1]
 [1 1]]
```

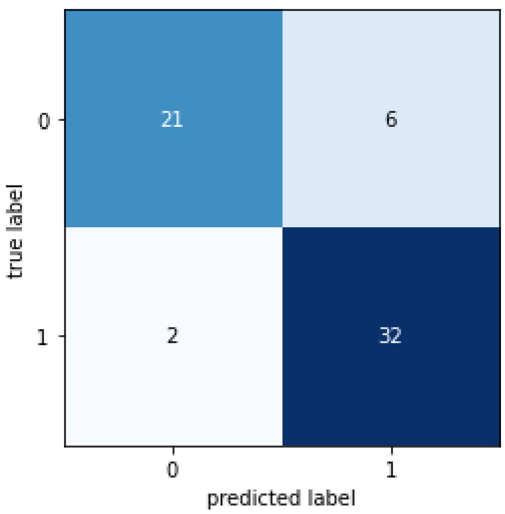
In [31]:

```
# printing the accuracy score and the confusion matrix
from sklearn.metrics import accuracy_score,confusion_matrix
acc = accuracy_score(y_te,y_pred)
cm = confusion_matrix(y_te,y_pred)
print(acc)
print(cm)
```

```
0.8688524590163934
[[21  6]
 [ 2 32]]
```

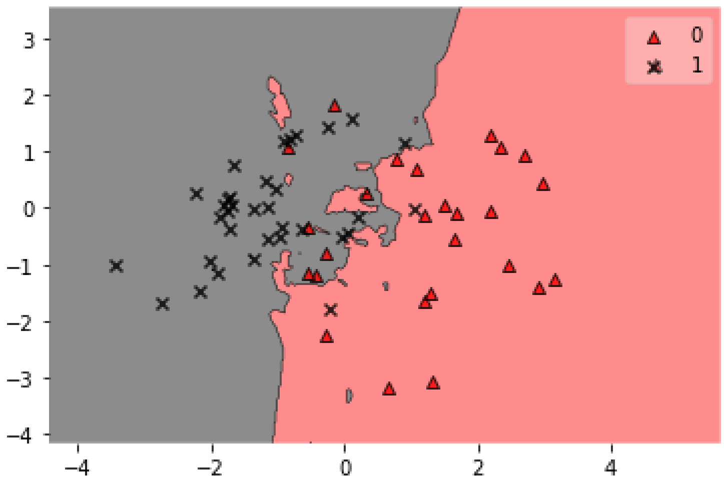
In [32]:

```
# plotting the confusion matrix and the decision regions
from mlxtend.plotting import plot_confusion_matrix,plot_decision_regions
plot_confusion_matrix(cm,cmap = 'Blues')
plt.show()
```



In [33]:

```
plot_decision_regions(X=x_te,y=y_te,clf=classifier,colors='red,black',markers='^x')
plt.show()
```



In [ ]: