In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
data=pd.read_csv('D:\heart.csv')
data
```

Out[2]:

| | Age | Sex | ChestPainType | RestingBP | Cholesterol | FastingBS | RestingECG | MaxHR | ExerciseA |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 40 | M | ATA | 140 | 289 | 0 | Normal | 172 | |
| **1** | 49 | F | NAP | 160 | 180 | 0 | Normal | 156 | |
| **2** | 37 | M | ATA | 130 | 283 | 0 | ST | 98 | |
| **3** | 48 | F | ASY | 138 | 214 | 0 | Normal | 108 | |
| **4** | 54 | M | NAP | 150 | 195 | 0 | Normal | 122 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **913** | 45 | M | TA | 110 | 264 | 0 | Normal | 132 | |
| **914** | 68 | M | ASY | 144 | 193 | 1 | Normal | 141 | |
| **915** | 57 | M | ASY | 130 | 131 | 0 | Normal | 115 | |
| **916** | 57 | F | ATA | 130 | 236 | 0 | LVH | 174 | |
| **917** | 38 | M | NAP | 138 | 175 | 0 | Normal | 173 | |

918 rows × 12 columns

In [3]:
```
data.dtypes
```

Out[3]:
```
Age                int64
Sex               object
ChestPainType     object
RestingBP          int64
Cholesterol        int64
FastingBS          int64
RestingECG        object
MaxHR              int64
ExerciseAngina    object
Oldpeak          float64
ST_Slope          object
HeartDisease       int64
dtype: object
```

In [4]:
```
data.isnull().sum()
```

Out[4]:
```
Age              0
Sex              0
ChestPainType    0
RestingBP        0
Cholesterol      0
FastingBS        0
RestingECG       0
MaxHR            0
ExerciseAngina   0
Oldpeak          0
ST_Slope         0
HeartDisease     0
dtype: int64
```

In [5]:
```
x =data.iloc[:,0:11]
x
```

Out[5]:

| | Age | Sex | ChestPainType | RestingBP | Cholesterol | FastingBS | RestingECG | MaxHR | ExerciseA |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 40 | M | ATA | 140 | 289 | 0 | Normal | 172 | |
| **1** | 49 | F | NAP | 160 | 180 | 0 | Normal | 156 | |
| **2** | 37 | M | ATA | 130 | 283 | 0 | ST | 98 | |
| **3** | 48 | F | ASY | 138 | 214 | 0 | Normal | 108 | |
| **4** | 54 | M | NAP | 150 | 195 | 0 | Normal | 122 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **913** | 45 | M | TA | 110 | 264 | 0 | Normal | 132 | |
| **914** | 68 | M | ASY | 144 | 193 | 1 | Normal | 141 | |
| **915** | 57 | M | ASY | 130 | 131 | 0 | Normal | 115 | |
| **916** | 57 | F | ATA | 130 | 236 | 0 | LVH | 174 | |
| **917** | 38 | M | NAP | 138 | 175 | 0 | Normal | 173 | |

918 rows × 11 columns

In [6]:
```
y=data['HeartDisease']
y
```

Out[6]:
```
0      0
1      1
2      0
3      1
4      0
      ..
913    1
914    1
915    1
916    1
917    0
Name: HeartDisease, Length: 918, dtype: int64
```

In [7]:
```
data.Age.unique()
```

Out[7]:
```
array([40, 49, 37, 48, 54, 39, 45, 58, 42, 38, 43, 60, 36, 44, 53, 52, 51,
       56, 41, 32, 65, 35, 59, 50, 47, 31, 46, 57, 55, 63, 66, 34, 33, 61,
       29, 62, 28, 30, 74, 68, 72, 64, 69, 67, 73, 70, 77, 75, 76, 71],
      dtype=int64)
```

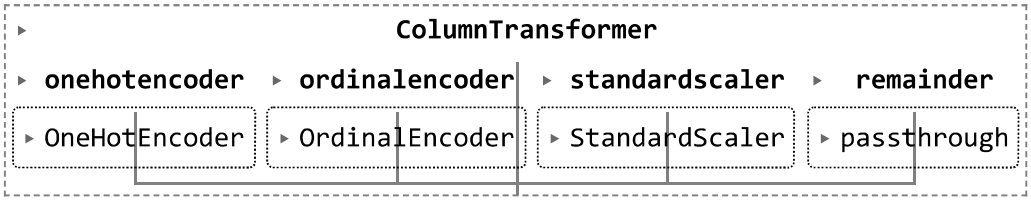
In [8]:
```
data.iloc[:,11].unique()
```

Out[8]:
```
array([0, 1], dtype=int64)
```

In [9]:
```
nomi_col=[2,6,10]
ordi_col=[1,8]
num_col=[0,3,4,5,7,9]
```

In [10]:
```
from sklearn.preprocessing import OneHotEncoder,OrdinalEncoder,StandardScaler
from sklearn.compose import make_column_transformer
from sklearn import set_config
trans = make_column_transformer((OneHotEncoder(sparse=False),nomi_col),
                                (OrdinalEncoder(),ordi_col),
                                (StandardScaler(),num_col),
                                 remainder='passthrough')
set_config(display='diagram')
trans
```

Out[10]:



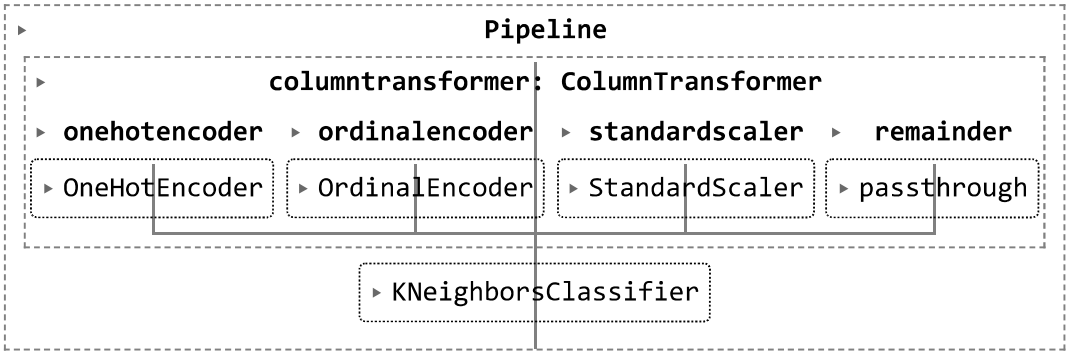

In [11]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size= 0.2)
```

In [12]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.pipeline import make_pipeline
Model= KNeighborsClassifier(10)
pipe = make_pipeline(trans, Model)
```

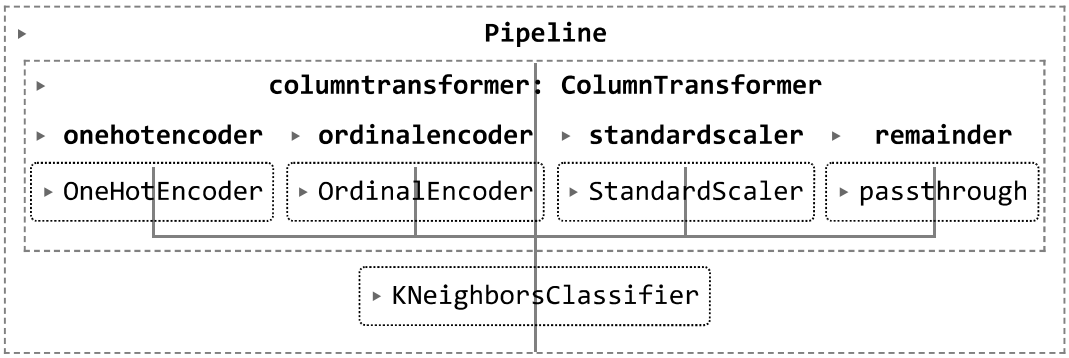
In [13]:
```
Model
```

Out[13]:
```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=10)
```

In [14]:
```
pipe
```

Out[14]:



In [15]:
```
pipe.fit(x_train, y_train)
```

Out[15]:



In [16]:
```
pred=pipe.predict (x_test)
```

In [17]:
```
pred
```

Out[17]:
```
array([1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1,
       0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0,
       1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1,
       1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0,
       0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1,
       1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0,
       1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1,
       1, 0, 1, 1, 0, 0, 0, 0], dtype=int64)
```

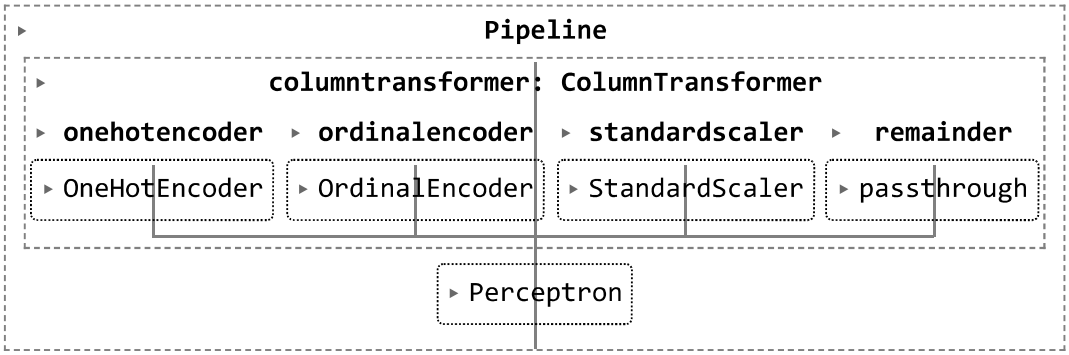
In [18]:
```
from sklearn.metrics import accuracy_score
accuracy_score(pred, y_test)*100
```

Out[18]:
```
85.86956521739131
```

In [19]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.pipeline import make_pipeline
perceptronalg= KNeighborsClassifier(10)
pipe = make_pipeline(trans, perceptronalg)
```

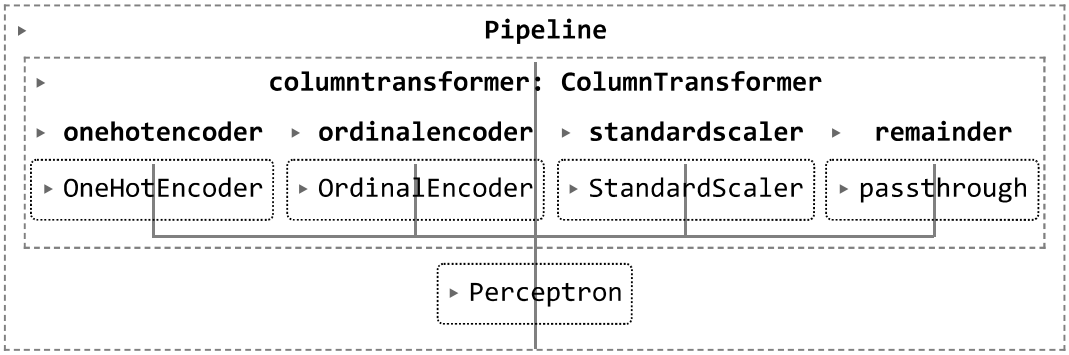
In [20]:
```
from sklearn.linear_model import Perceptron
from sklearn.pipeline import make_pipeline
prc = Perceptron(class_weight='balanced')
#perceptronalg= KNeighborsClassifier(2)
pipe_prc = make_pipeline(trans,prc)
pipe_prc
```

Out[20]:



In [21]:
```
pipe_prc.fit(x_train,y_train)
```

Out[21]:




In [22]:
```
pred_prc= pipe_prc.predict(x_test)
```

In [23]:
```
accuracy_score(pred_prc, y_test)*100
```

Out[23]:
```
76.08695652173914
```

In [24]:
```
x2=data.loc[:,['MaxHR','Age']]
y2=data.HeartDisease
#Model= KNeighborsClassifier(10)
prc = Perceptron(class_weight='balanced')

prc.fit(x2,y2)
```

Out[24]:
```
Perceptron(class_weight='balanced')
```

In [25]:
```
pip install mlxtend
```

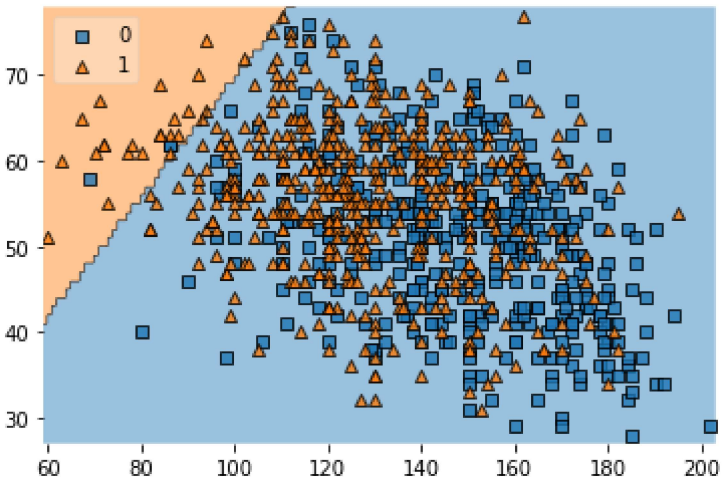
```
Requirement already satisfied: mlxtend in c:\users\viplo\anaconda3\lib\site-pac
kages (0.19.0)
Requirement already satisfied: scipy>=1.2.1 in c:\users\viplo\anaconda3\lib\sit
e-packages (from mlxtend) (1.6.2)
Requirement already satisfied: setuptools in c:\users\viplo\anaconda3\lib\site-
packages (from mlxtend) (52.0.0.post20210125)
Requirement already satisfied: scikit-learn>=0.20.3 in c:\users\viplo\anaconda3
\lib\site-packages (from mlxtend) (1.0.2)
Requirement already satisfied: pandas>=0.24.2 in c:\users\viplo\anaconda3\lib\s
ite-packages (from mlxtend) (1.2.4)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\viplo\anaconda3\li
b\site-packages (from mlxtend) (3.3.4)
Requirement already satisfied: numpy>=1.16.2 in c:\users\viplo\anaconda3\lib\si
te-packages (from mlxtend) (1.19.5)
Requirement already satisfied: joblib>=0.13.2 in c:\users\viplo\anaconda3\lib\s
ite-packages (from mlxtend) (1.0.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\u
sers\viplo\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.4.
7)
Requirement already satisfied: pillow>=6.2.0 in c:\users\viplo\anaconda3\lib\si
te-packages (from matplotlib>=3.0.0->mlxtend) (8.2.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\viplo\anaconda3\li
b\site-packages (from matplotlib>=3.0.0->mlxtend) (1.3.1)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\viplo\anaconda3
\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.8.1)
Requirement already satisfied: cycler>=0.10 in c:\users\viplo\anaconda3\lib\sit
e-packages (from matplotlib>=3.0.0->mlxtend) (0.10.0)
Requirement already satisfied: six in c:\users\viplo\anaconda3\lib\site-package
s (from cycler>=0.10->matplotlib>=3.0.0->mlxtend) (1.15.0)
Requirement already satisfied: pytz>=2017.3 in c:\users\viplo\anaconda3\lib\sit
e-packages (from pandas>=0.24.2->mlxtend) (2021.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\viplo\anaconda3
\lib\site-packages (from scikit-learn>=0.20.3->mlxtend) (2.1.0)
Note: you may need to restart the kernel to use updated packages.
```

In [26]:
```
import mlxtend
```

In [27]:
```
from mlxtend.plotting import plot_decision_regions
plot_decision_regions(x2.values,y2.values,clf=prc,legend=2)
```

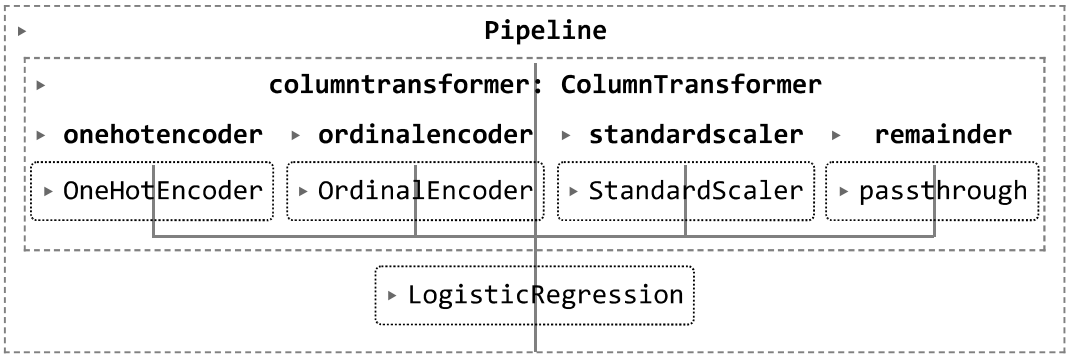
```
C:\Users\viplo\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X
does not have valid feature names, but Perceptron was fitted with feature names
  warnings.warn(
```

Out[27]: <AxesSubplot:>



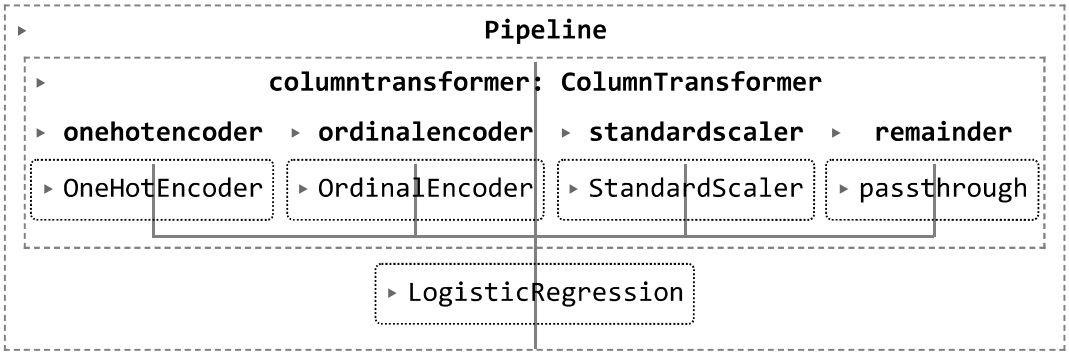

In [28]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import make_pipeline
model= LogisticRegression (solver='liblinear')
pipe= make_pipeline(trans,model)
pipe
```

Out[28]:



In [29]:
```
pipe.fit(x_train,y_train)
```

Out[29]:




In [30]:
```
prediction = pipe.predict(x_test)
```

In [31]:
```
prediction
```

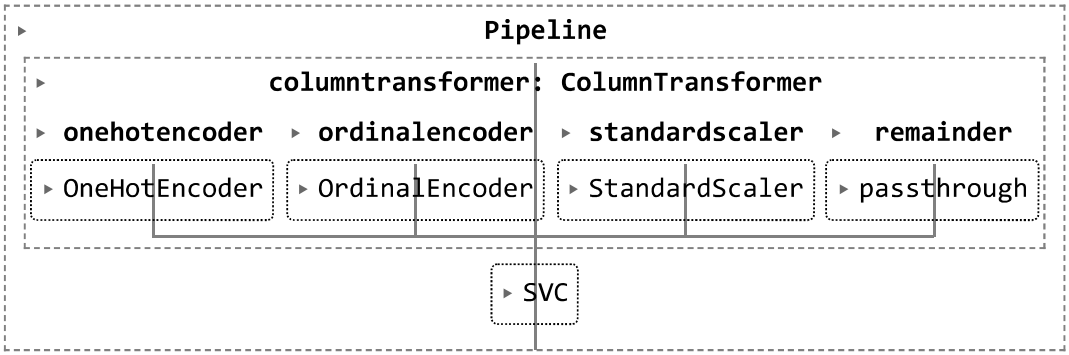
Out[31]:
```
array([1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0,
       1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0,
       0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1,
       1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0,
       1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1,
       1, 1, 1, 1, 0, 0, 1, 0], dtype=int64)
```

In [32]:
```
accuracy_score(prediction,y_test)*100
```

Out[32]:
```
84.78260869565217
```

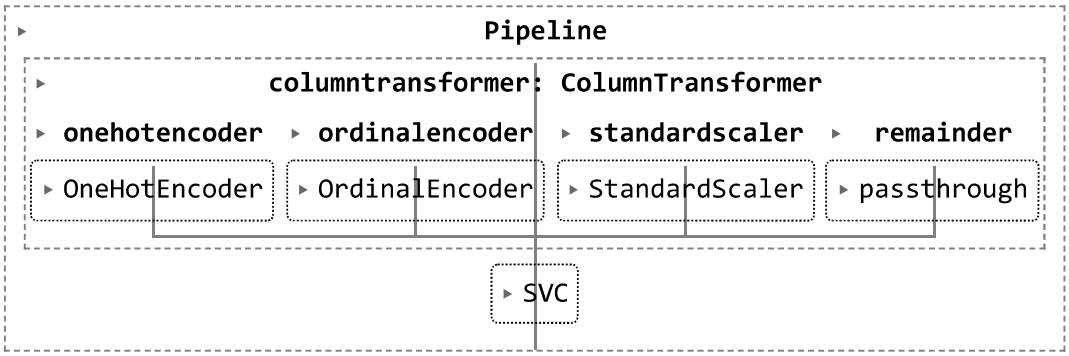
In [33]:
```
from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline
algorithm=SVC()
pipe3= make_pipeline(trans,algorithm)
pipe3
```

Out[33]:




In [34]:
```
pipe3.fit(x_train,y_train)
```

Out[34]:




In [35]:
```
prediction3=pipe3.predict(x_test)
```

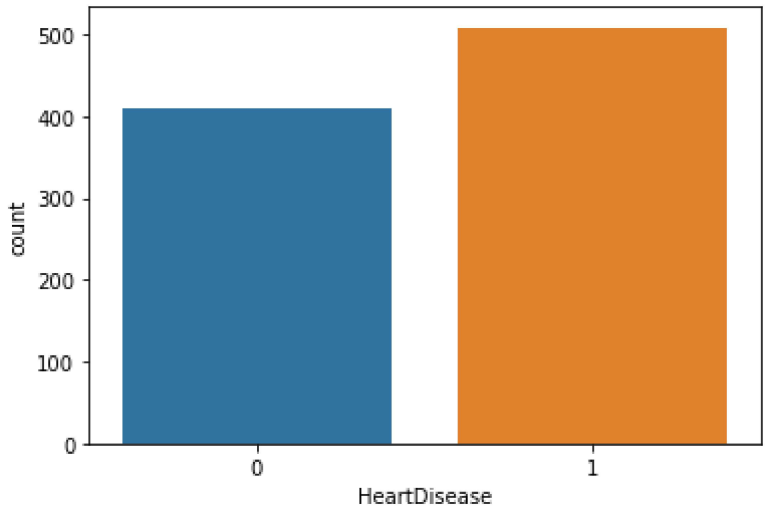
In [36]:
```
accuracy_score(prediction3,y_test)*100
```

Out[36]:
```
84.23913043478261
```

In [37]:
```
import seaborn as sns
sns.countplot(y)
```

```
C:\Users\viplo\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variable as a keyword arg: x. From version 0.12, the o
nly valid positional argument will be `data`, and passing other arguments witho
ut an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[37]: <AxesSubplot:xlabel='HeartDisease', ylabel='count'>




In [40]:
```
pip install imblearn
```

```
Collecting imblearnNote: you may need to restart the kernel to use updated pack
ages.
  Using cached imblearn-0.0-py2.py3-none-any.whl (1.9 kB)
Collecting imbalanced-learn
  Using cached imbalanced_learn-0.9.0-py3-none-any.whl (199 kB)
Requirement already satisfied: scikit-learn>=1.0.1 in c:\users\viplo\anaconda3
\lib\site-packages (from imbalanced-learn->imblearn) (1.0.2)
Requirement already satisfied: numpy>=1.14.6 in c:\users\viplo\anaconda3\lib\si
te-packages (from imbalanced-learn->imblearn) (1.19.5)
Requirement already satisfied: scipy>=1.1.0 in c:\users\viplo\anaconda3\lib\sit
e-packages (from imbalanced-learn->imblearn) (1.6.2)
Requirement already satisfied: joblib>=0.11 in c:\users\viplo\anaconda3\lib\sit
e-packages (from imbalanced-learn->imblearn) (1.0.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\viplo\anaconda3
\lib\site-packages (from imbalanced-learn->imblearn) (2.1.0)
Installing collected packages: imbalanced-learn, imblearn
Successfully installed imbalanced-learn-0.9.0 imblearn-0.0
```

In [41]:
```
import imblearn
```

In [42]:
```
from imblearn.under_sampling import RandomUnderSampler
under= RandomUnderSampler()
u_x,u_y = under.fit_resample(x,y)
u_y.value_counts()
```

Out[42]:
```
0    410
1    410
Name: HeartDisease, dtype: int64
```

In [43]:
```
from imblearn.over_sampling import RandomOverSampler
under = RandomOverSampler()
u_x, u_y = under.fit_resample(x,y)
u_y.value_counts()
```

Out[43]:
```
0    508
1    508
Name: HeartDisease, dtype: int64
```

In [44]:
```
from sklearn.metrics import accuracy_score
accuracy_score(pred, y_test)*100
```

Out[44]:
```
85.86956521739131
```