

```
import pandas as pd = pd.read_csv('/content/heart_disease_uci.csv')
```

```
df.head()
```

	id	sex
0	1	Male
1	2	Male
2	3	Male
3	4	Male
4	5	Female

```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
for col in df.select_dtypes(include=['object']).columns:
    df[col] = encoder.fit_transform(df[col])
```

```
x = df.iloc[:,0]
y = df.iloc[:,1]
```

```
import numpy as np
x = np.array(x)
y = np.array(y)
```

```
x = x.reshape(-1,1)
y = y.reshape(-1,1)
```

```
from sklearn.model_selection import train_test_split
```

```
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, random_state=42
)
print("Training sample:", x_train.shape)
print("testing sample:", x_test.shape)
```

```
Training sample: (736, 1)
testing sample: (184, 1)
```

```
from sklearn.svm import SVC
```

```
svm_liner = SVC(kernel='RBF', degree=2, class_weight='balanced')
```

```
svm_liner = SVC(kernel='rbf', degree=2, class_weight='balanced', gamma=0.1)
svm_liner.fit(x_train, y_train.ravel())
```

SVC
 ⓘ ⓘ

```
SVC(class_weight='balanced', degree=2, gamma=0.1)
```

```
y_pred=svm_liner.predict(x_test)
```

```
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score, f1_score
```

```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy : ", accuracy)
```

```
Accuracy : 0.6413043478260869
```

```
cm = confusion_matrix(y_test, y_pred, labels=[1,0])
print("Confusion Matrix :")
print("TP FP")
print("FN TN")
print(cm)
```

```
Confusion Matrix :  
TP FP  
FN TN  
[[95 45]  
[21 23]]
```

```
precision = precision_score(y_test,y_pred,zero_division=0)  
print("Precision :",precision)
```

```
Precision : 0.8189655172413793
```

```
f1 = f1_score(y_test, y_pred, zero_division=0)  
print("F1 Score:",f1)
```

```
F1 Score: 0.7421875
```

```
from sklearn.model_selection import GridSearchCV  
print("GridSearchCV imported successfully.")
```

```
GridSearchCV imported successfully.
```

```
param_grid = {'gamma': [0.2, 0.3, 0.4, 1], 'C': [0.1, 0.2, 0.3, 0.4, 0.5, 1]}  
print("Parameter grid defined successfully.")
```

```
Parameter grid defined successfully.
```

```
svc_model = SVC(kernel='rbf', class_weight='balanced')  
grid_search = GridSearchCV(svc_model, param_grid, cv=5)  
grid_search.fit(x_train, y_train.to_numpy())  
print("Grid search completed.")
```

```
Grid search completed.
```

```
Start coding or generate with AI.
```

```
print("Best parameters:", grid_search.best_params_)  
print("Best cross-validation score:", grid_search.best_score_)
```

```
Best parameters: {'C': 0.2, 'gamma': 0.3}  
Best cross-validation score: 0.8111877183305755
```

```
print("F1 Score:", f1)
```

```
F1 Score: 0.7421875
```

```
Start coding or generate with AI.
```