

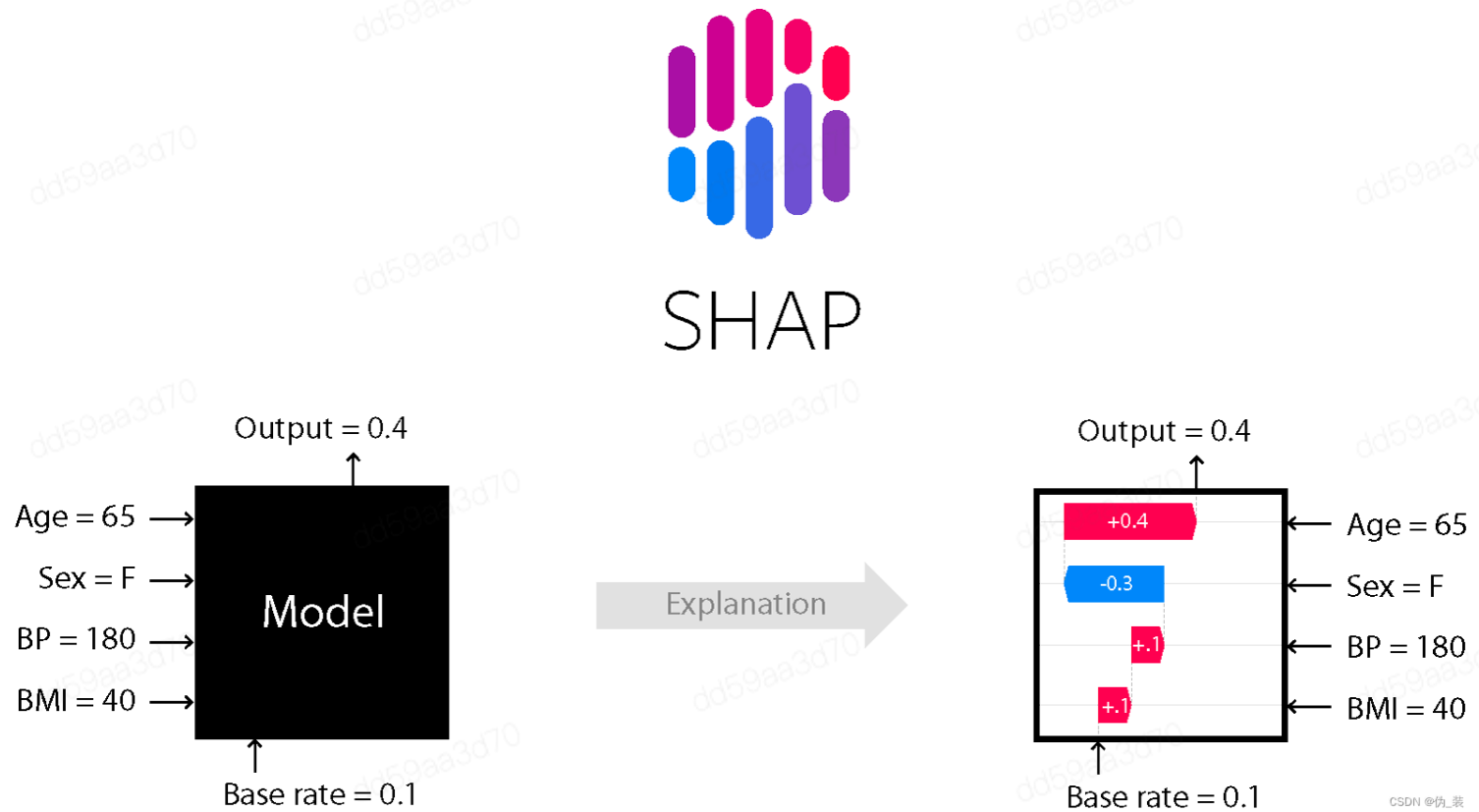
# 可解释性工具

## 目录

- 1.摘要
- 2.业务概述
  - 2.1 业务视图
  - 2.2 功能描述
- 3.需求实现
  - 安装
  - 构建数据集、训练模型、对测试集样本预测
  - 计算shap值
  - 特征重要度
  - 各特征的数值大小与各特征的shap值关系图

## 1.摘要

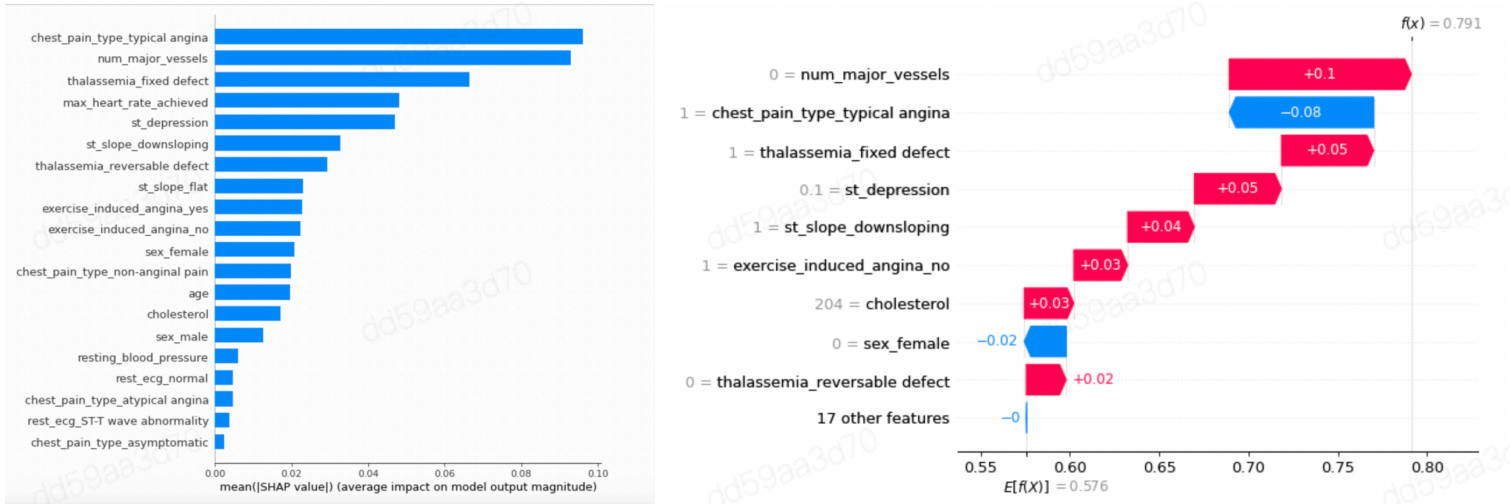
在执行分类任务时，可通过SHAP库工具打破深度学习黑箱子，了解模型学习过程。



## 2.业务概述

### 2.1 业务视图

SHAP（SHapley Additive exPlanations）是一种用于解释预测结果的方法，它基于Shapley值理论，通过将预测结果分解为每个特征的影响，为模型提供全局和局部的可解释性。



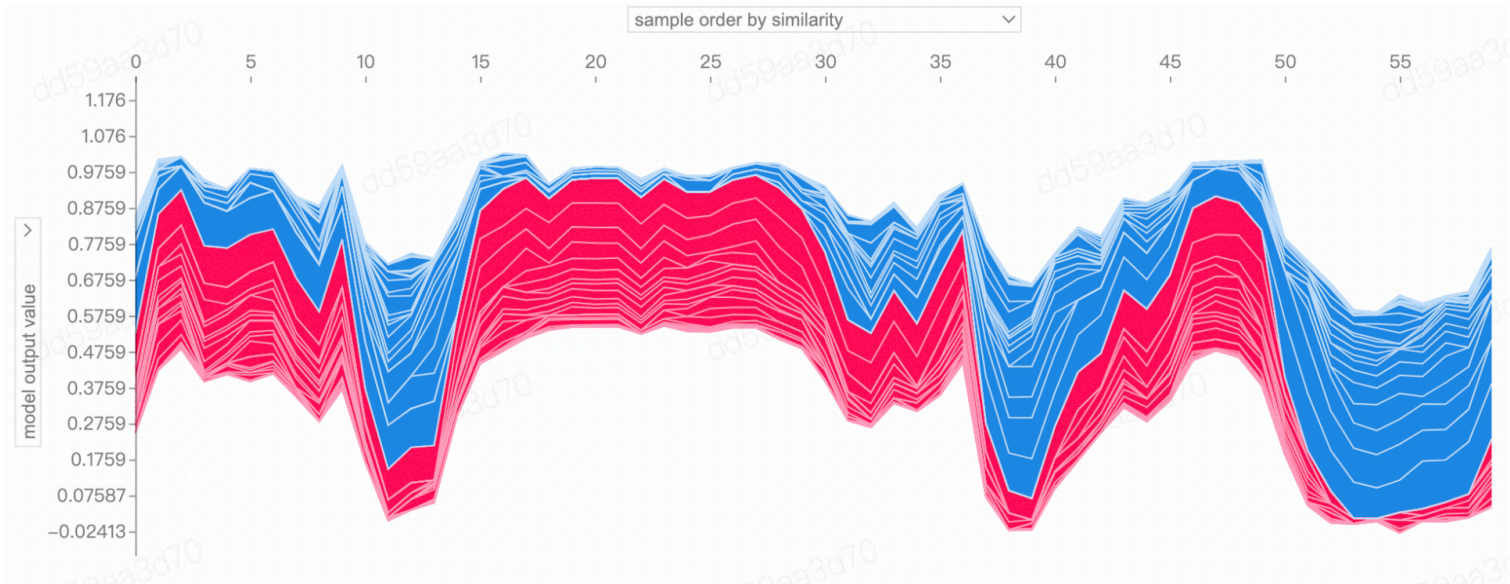
本节简要描述从系统外部看到的系统的特征，比如谁使用该系统解决什么业务问题，该业务问题原先是如何解决的，该系统是如何改进的。

注意：

一般小项目简要描述即可，对于比较复杂的大项目，该节需要单独再列出小标题详细说明。

### 2.2 功能描述

SHAP的核心思想是将特征值的贡献分配到不同的特征中，计算每个特征的Shapley值，并将其与特征值相乘得到该特征对于预测结果的贡献，SHAP可以用于机器学习模型，包括分类和回归模型，可以生成图像化和定量的解释结果，帮助用户解释模型的决策过程。通过shap工具可以计算出各个特征对于模型分类的结果贡献度，通过贡献度值的大小可以筛选模型重点关注的对象以及决策依据。



提示：(必写)

以文字配合图示（可以为用例图）简要说明软件的主要功能、输入输出等内容；  
如果使用用例图，则需要对比较复杂的重要用例进行再说明，描述其主要过程。

### 3.需求实现

本节使用shap机器学习可解释性分析工具包，对uci心脏病数据集上训练得到的随机森林分类模型进行可解释性分析。得到特征重要性图、shap值图、单个样本所有特征的shap值图、某一特征不同值的shap值图。赋予机器学习模型可解释性，打破机器学习模型的黑箱子，让我们能更加充分了解模型、改进模型、信任模型，并对每一个数据样本构建精准定制的数学模型。

### 安装

</> Bash

```
1 pip install shap
2 #
3 conda install -c conda-forge shap
```

### 构建数据集、训练模型、对测试集样本预测

</> Python

```
1 import numpy as np
2 import pandas as pd
3
4 import matplotlib.pyplot as plt
5 %matplotlib inline
6
7 # 忽略烦人的红色提示
8 import warnings
9 warnings.filterwarnings("ignore")
10
11 # 忽略烦人的红色提示
12 import warnings
13 warnings.filterwarnings("ignore")
14
15 # 导入数据集，划分特征和标签
16 df = pd.read_csv('process_heart.csv')
17 X = df.drop('target',axis=1)
18 y = df['target']
```

```
19
20 # 划分训练集和测试集
21 from sklearn.model_selection import train_test_split
22 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
23                                                    random_state=10)
24
25 # 构建随机森林模型
26 from sklearn.ensemble import RandomForestClassifier
27 model = RandomForestClassifier(max_depth=5, n_estimators=100)
28 model.fit(X_train, y_train)
29
30 y_pred = model.predict(X_test)
31
32 y_pred_proba = model.predict_proba(X_test)
```

## 计算shap值

&lt;/&gt;

Python

```
1 import shap
2 shap.initjs()
3
4 explainer = shap.TreeExplainer(model)
5 shap_values = explainer.shap_values(X_test)
```

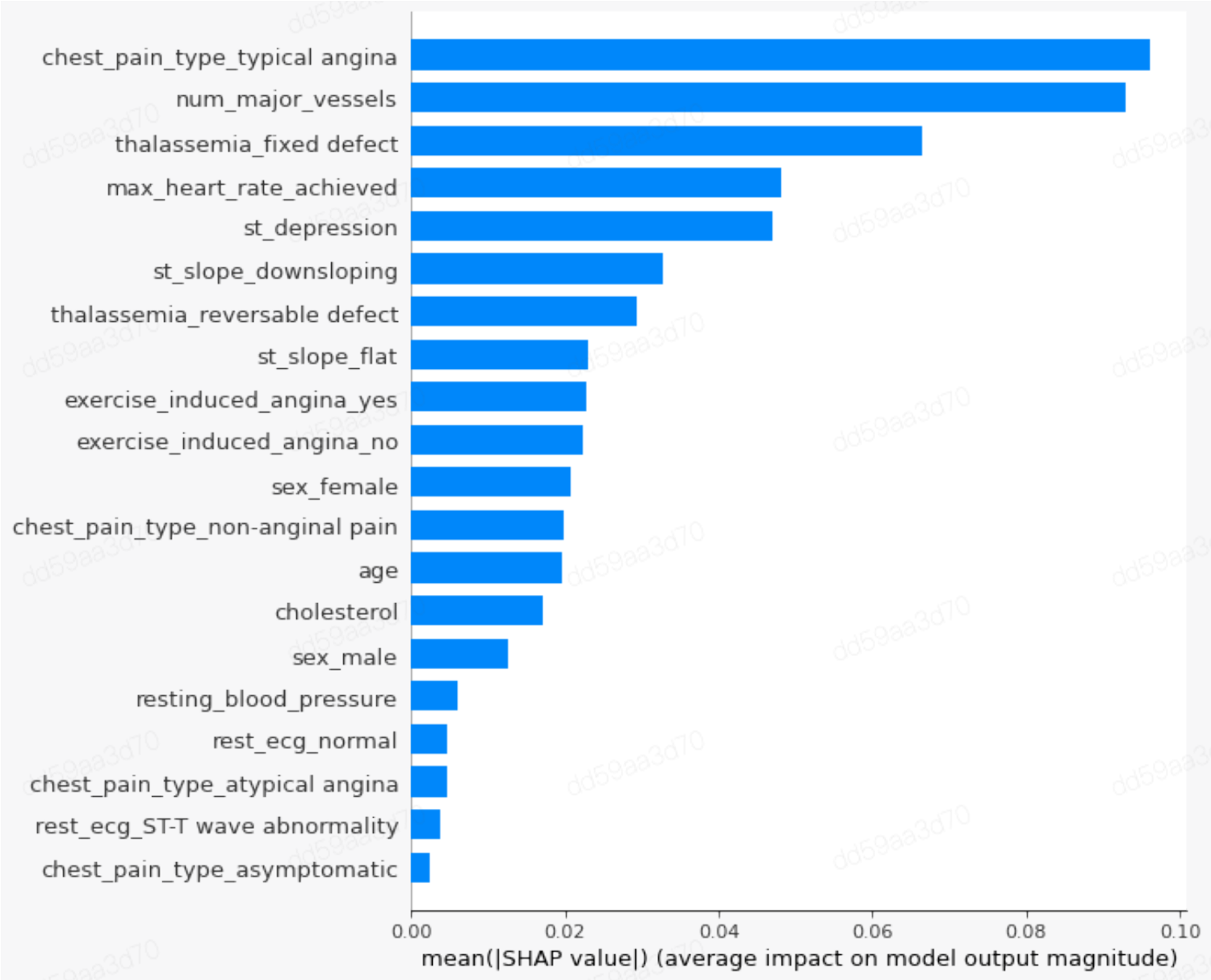
## 特征重要度

对于某个特征，计算测试集每个病人的该特征shap值之和，shap值越高，特征越重要。

&lt;/&gt;

Python

```
1 shap.summary_plot(shap_values[1], X_test, plot_type="bar")
```

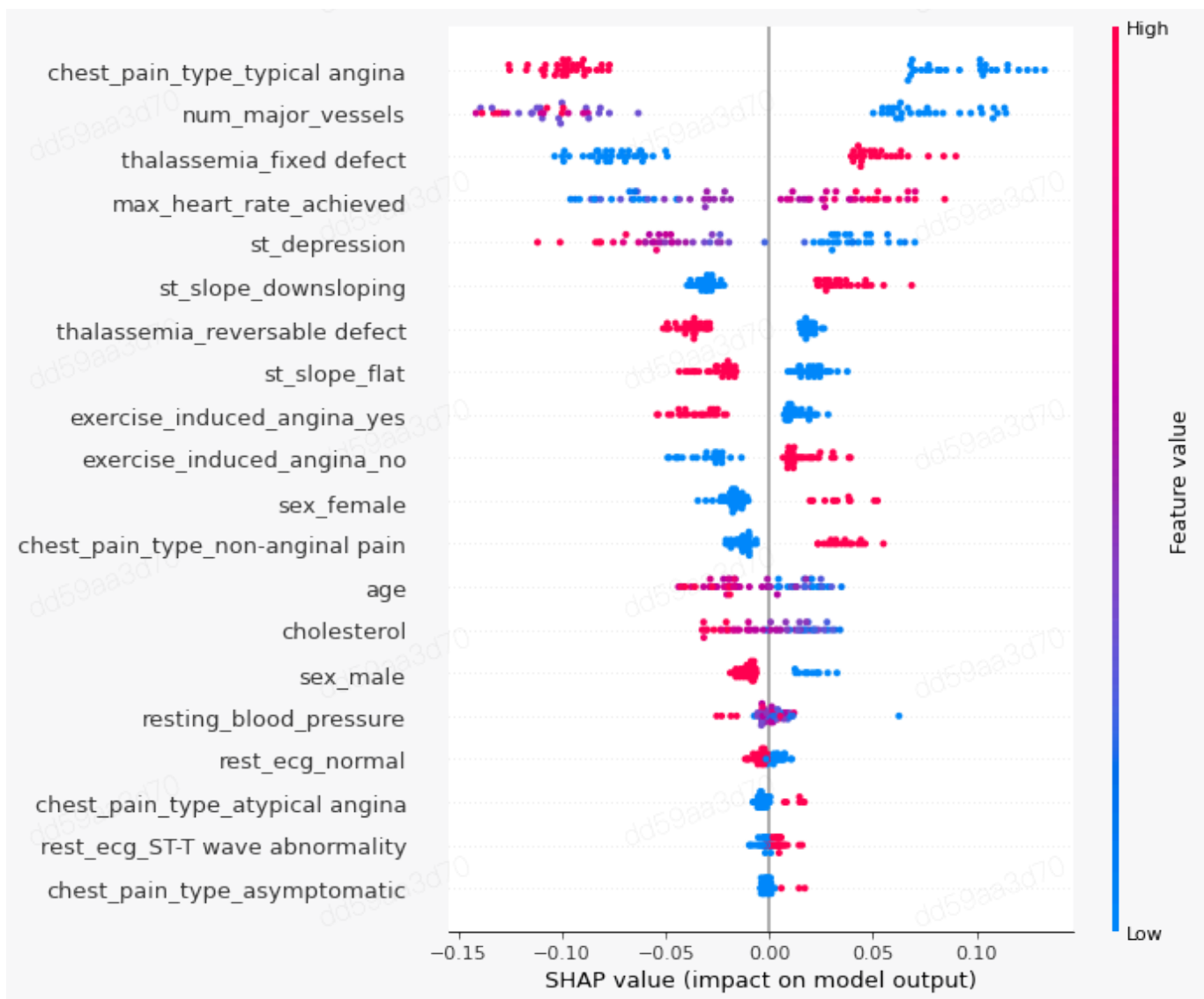


## 各特征的数值大小与各特征的shap值关系图

每一行表示一个特征，红色表示该特征的值较高的数据点，蓝色值表示该特征的值较低的数据点。越靠右的点表示这个特征对预测为“患病”的正向影响越高。

In [4]:

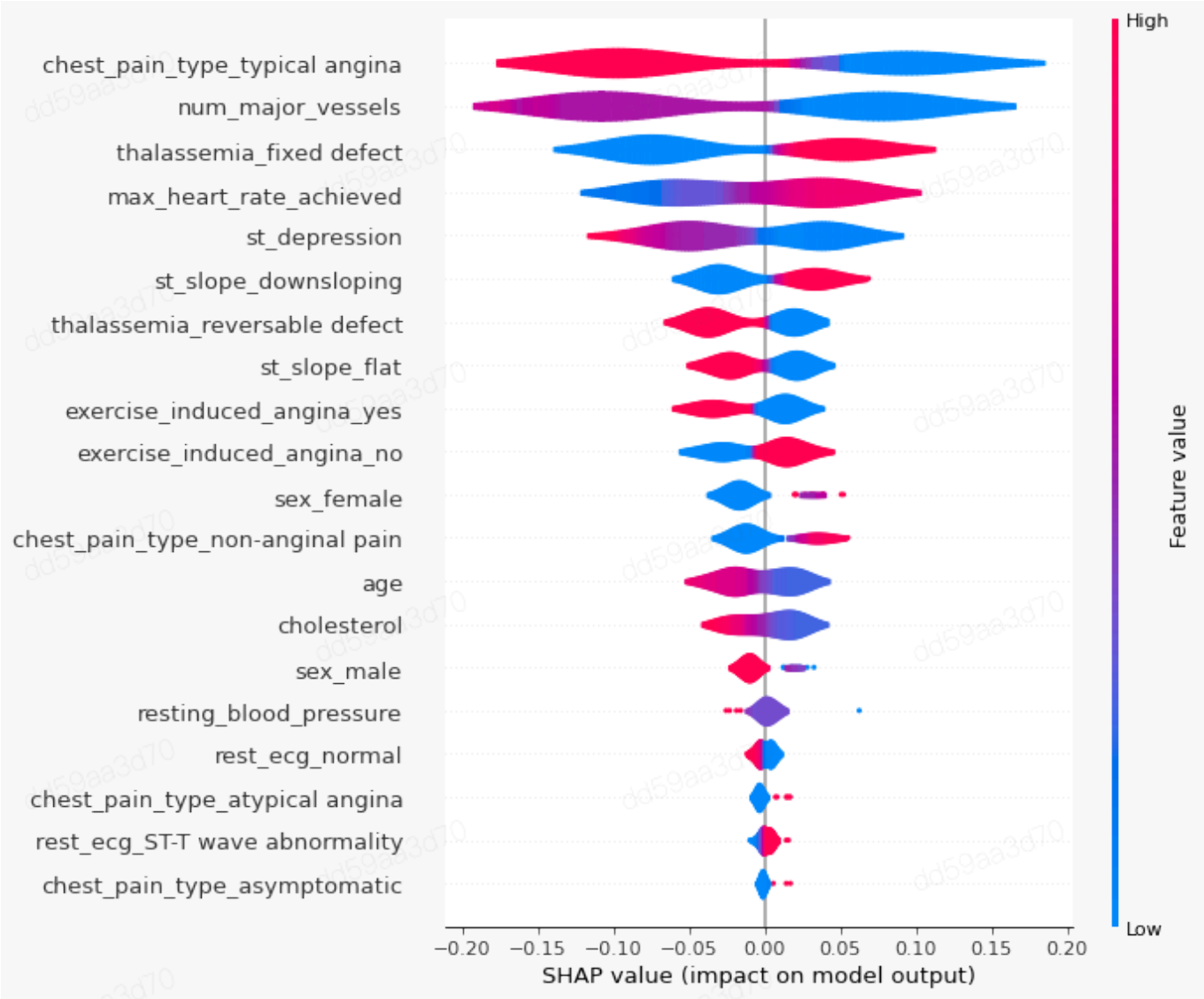
```
shap.summary_plot(shap_values[1],X_test)
```



In [5]:

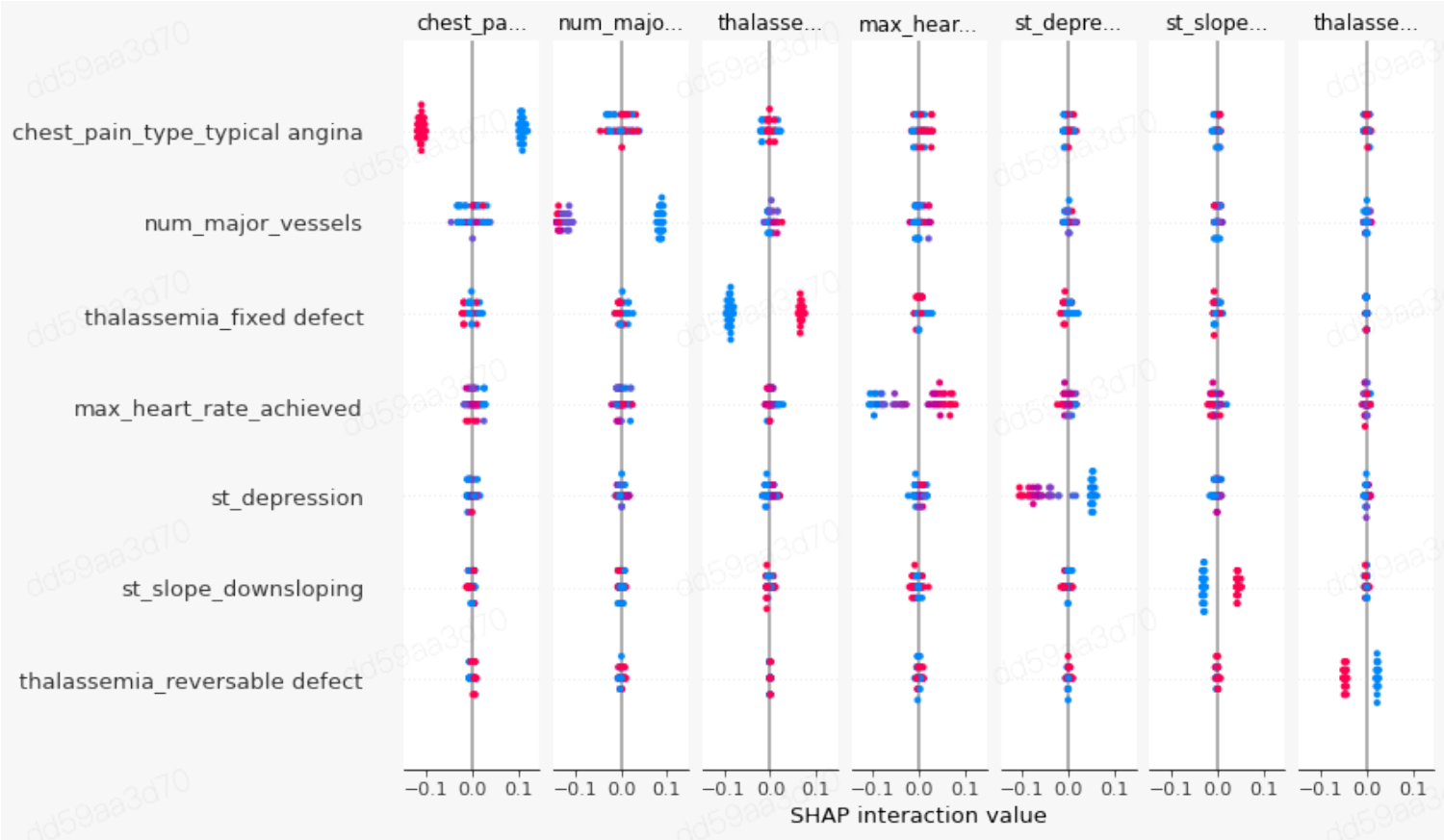
```
shap.summary_plot(shap_values[1],X_test,plot_type="violin")
```





num\_major\_vessels越大，对预测为“患病”结果的正向贡献越小。  
max\_heart\_rate\_achieved越大，对预测为“患病”结果的正向贡献越大。

```
In [6]:  
shap_interaction_values=explainer.shap_interaction_values(X_test)shap.summary_plot(shap_interaction_v  
alues[1],X_test)
```

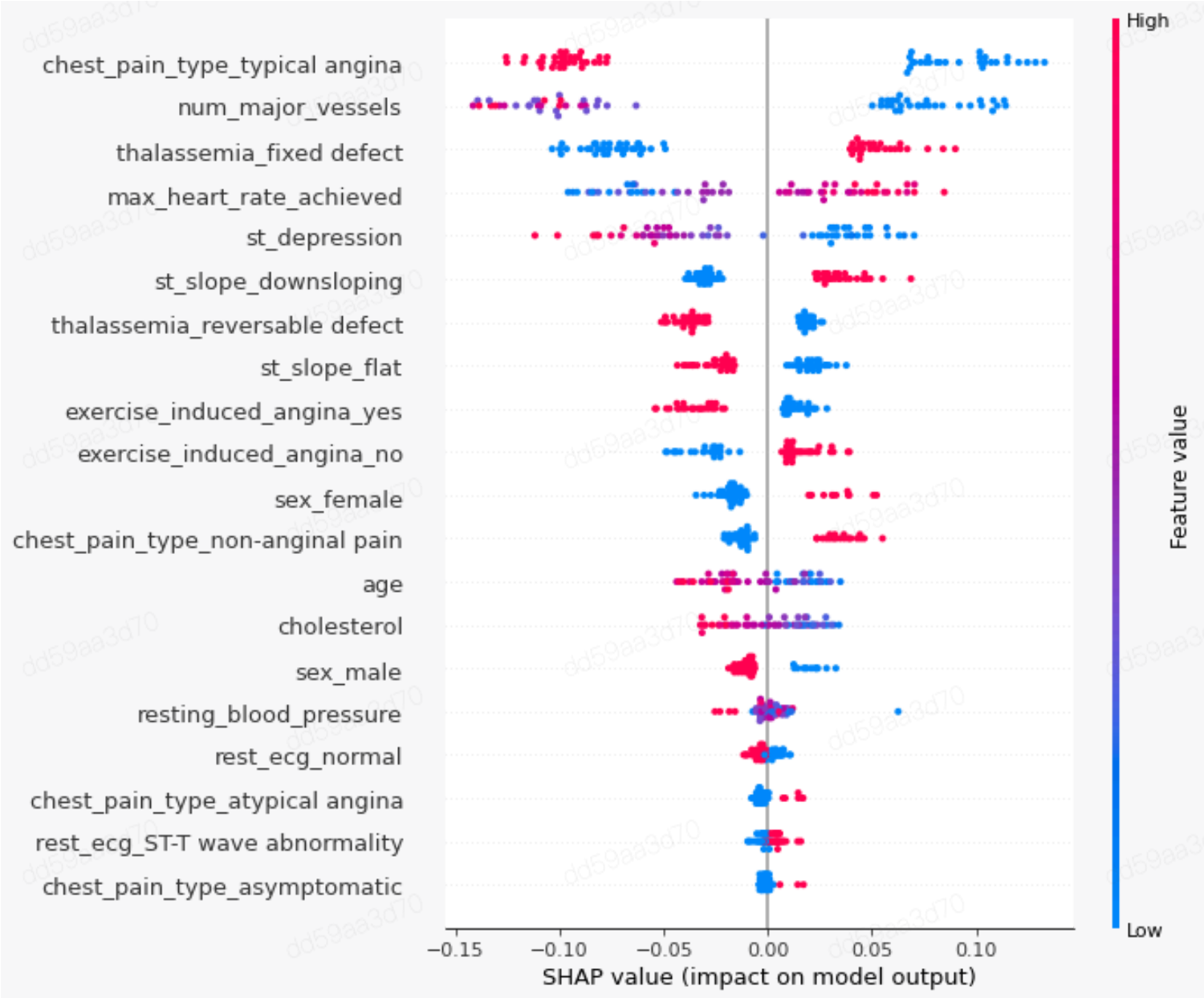


每一行表示一个特征，红色表示该特征的值较高的数据点，蓝色值表示该特征的值较低的数据点。  
越靠右的点表示这个特征对预测为“患病”的正向影响越高。

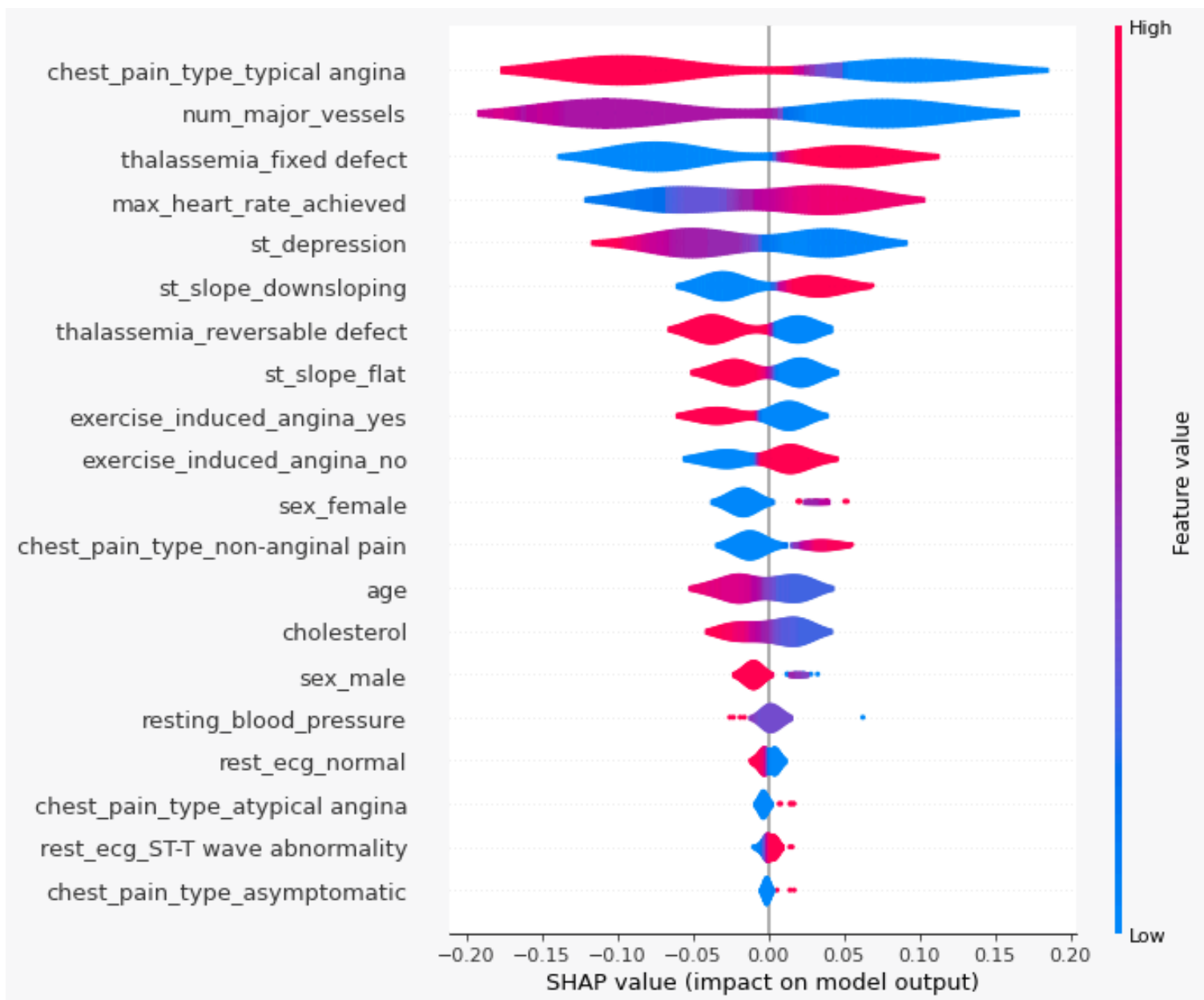
</> Python

```
1 shap.summary_plot(shap_values[1], X_test)
```





```
</> Python
1 shap.summary_plot(shap_values[1], X_test, plot_type="violin")
```



num\_major\_vessels越大，对预测为“患病”结果的正向贡献越小。

max\_heart\_rate\_achieved越大，对预测为“患病”结果的正向贡献越大。

&lt;/&gt;

Python

```
1 shap_interaction_values=explainer.shap_interaction_values(X_test)shap.summary_plot(
  shap_interaction_values[1],X_test)
```

