



What Causes Heart Disease? Explaining the Model

Python notebook using data from [Heart Disease UCI](#) · 16,926 views · random forest, healthcare, binary classification, +1 more

<https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model>

1. Package Import

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns #for plotting
from sklearn.ensemble import RandomForestClassifier #for the model
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import export_graphviz #plot tree
from sklearn.metrics import roc_curve, auc #for model evaluation
from sklearn.metrics import classification_report #for model evaluation
from sklearn.metrics import confusion_matrix #for model evaluation
from sklearn.model_selection import train_test_split #for data splitting
import eli5 #for permutation importance
from eli5.sklearn import PermutationImportance
import shap #for SHAP values
from pdpbox import pdp, info_plots #for partial plots
np.random.seed(123) #ensure reproducibility

pd.options.mode.chained_assignment = None #hide any pandas warnings
```



What Causes Heart Disease? Explaining the Model

Python notebook using data from [Heart Disease UCI](#) · 16,926 views · random forest, healthcare, binary classification, +1 more

<https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model>

2. 데이터 확인 (총 : 303개)

dt.head(10)

	age	sex	cp	trestbps	chol	fbps	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
5	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
6	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
7	44	1	1	120	263	0	1	173	0	0.0	2	0	3	1
8	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
9	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1

It's a clean, easy to understand set of data. However, the meaning of some of the column headers are not obvious. Here's what they mean,

- **age:** The person's age in years
 - **sex:** The person's sex (1 = male, 0 = female)
 - **cp:** The chest pain experienced (Value 1: typical angina, Value 2: atypical angina, Value 3: non-anginal pain, Value 4: asymptomatic)
 - **trestbps:** The person's resting blood pressure (mm Hg on admission to the hospital)
 - **chol:** The person's cholesterol measurement in mg/dl
 - **fbps:** The person's fasting blood sugar (> 120 mg/dl, 1 = true; 0 = false)
 - **restecg:** Resting electrocardiographic measurement (0 = normal, 1 = having ST-T wave abnormality, 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria)
 - **thalach:** The person's maximum heart rate achieved
 - **exang:** Exercise induced angina (1 = yes; 0 = no)
 - **oldpeak:** ST depression induced by exercise relative to rest ('ST' relates to positions on the ECG plot. See more [here](#))
 - **slope:** the slope of the peak exercise ST segment (Value 1: upsloping, Value 2: flat, Value 3: downsloping)
 - **ca:** The number of major vessels (0-3)
 - **thal:** A blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect)
 - **target:** Heart disease (0 = no, 1 = yes)
-
- **age :** 나이
 - **sex :** 성별(1=남자, 0=여자)
 - **cp :** 가슴통증 경험 유무 (1 : 전형적인 협심증, 2 : 비정형 협심증, 3 : 비 협심증, 4 : 무증상)
 - **trestbps :** 혈압
 - **chol :** 콜레스테롤
 - **fbps :** 공복혈당
 - **restecg :** 심전도 (0=정상, 1=ST-T파 이상, 2 = Estes기준에 의한 확진 또는 명확한 좌심실 비대증 보여줌)
 - **thalach :** 최대 심박수
 - **exang :** 운동 유발 성 협심증 (1=예, 0=아니오)
 - **oldpeak :** 휴식과 관련된 운동으로 유발 된 ST 우울증
 - **slope :** 최고 운동 ST 세그먼트의 슬로프 (1:upsloping, 2 : flat, 3 : downsloping)
 - **ca :** 주요 혈관 수
 - **thal :** thalassemia라는 혈액 장애 (3=정상, 6 = 치료 받음, 7 = 고칠수 있는 상태)
 - **target :** 심장병 발생 여부(0=아니오, 1=예)



⌚ What Causes Heart Disease? Explaining the Model

Python notebook using data from [Heart Disease UCI](#) · 16,926 views · ♦ random forest, healthcare, binary classification, +1 more

<https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model>

3. 데이터 컬럼값 변환 (총 : 303개)

```
dt['sex'][dt['sex'] == 0] = 'female'
dt['sex'][dt['sex'] == 1] = 'male'

dt['chest_pain_type'][dt['chest_pain_type'] == 1] = 'typical angina'
dt['chest_pain_type'][dt['chest_pain_type'] == 2] = 'atypical angina'
dt['chest_pain_type'][dt['chest_pain_type'] == 3] = 'non-anginal pain'
dt['chest_pain_type'][dt['chest_pain_type'] == 4] = 'asymptomatic'

dt['fasting_blood_sugar'][dt['fasting_blood_sugar'] == 0] = 'lower than 120mg/ml'
dt['fasting_blood_sugar'][dt['fasting_blood_sugar'] == 1] = 'greater than 120mg/ml'

dt['rest_ecg'][dt['rest_ecg'] == 0] = 'normal'
dt['rest_ecg'][dt['rest_ecg'] == 1] = 'ST-T wave abnormality'
dt['rest_ecg'][dt['rest_ecg'] == 2] = 'left ventricular hypertrophy'

dt['exercise_induced_angina'][dt['exercise_induced_angina'] == 0] = 'no'
dt['exercise_induced_angina'][dt['exercise_induced_angina'] == 1] = 'yes'

dt['st_slope'][dt['st_slope'] == 1] = 'upsloping'
dt['st_slope'][dt['st_slope'] == 2] = 'flat'
dt['st_slope'][dt['st_slope'] == 3] = 'downsloping'

dt['thalassemia'][dt['thalassemia'] == 1] = 'normal'
dt['thalassemia'][dt['thalassemia'] == 2] = 'fixed defect'
dt['thalassemia'][dt['thalassemia'] == 3] = 'reversible defect'
```



What Causes Heart Disease? Explaining the Model

Python notebook using data from [Heart Disease UCI](#) · 16,926 views · random forest, healthcare, binary classification, +1 more

<https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model>

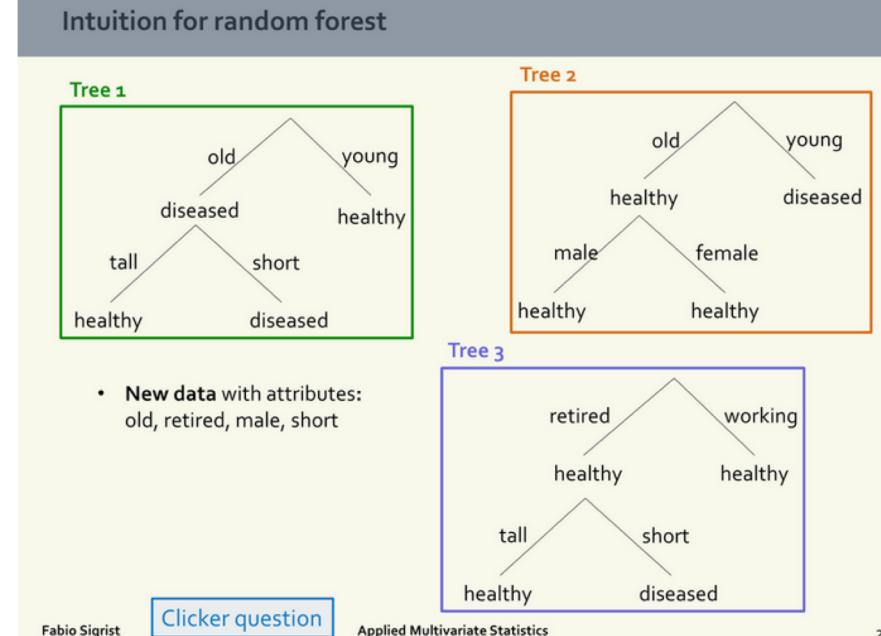
4. 학습 및 검증 확인 데이터베이스 생성 (학습 : 80%, 검증 : 20%)

```
X_train, X_test, y_train, y_test = train_test_split(dt.drop('target', 1), dt['target'], test_size = .2, random_state=10) #split the data
```

5. 학습 모델 생성 및 학습 [RandomForestClassifier, 5단계]

```
model = RandomForestClassifier(max_depth=5)
model.fit(X_train, y_train)
```

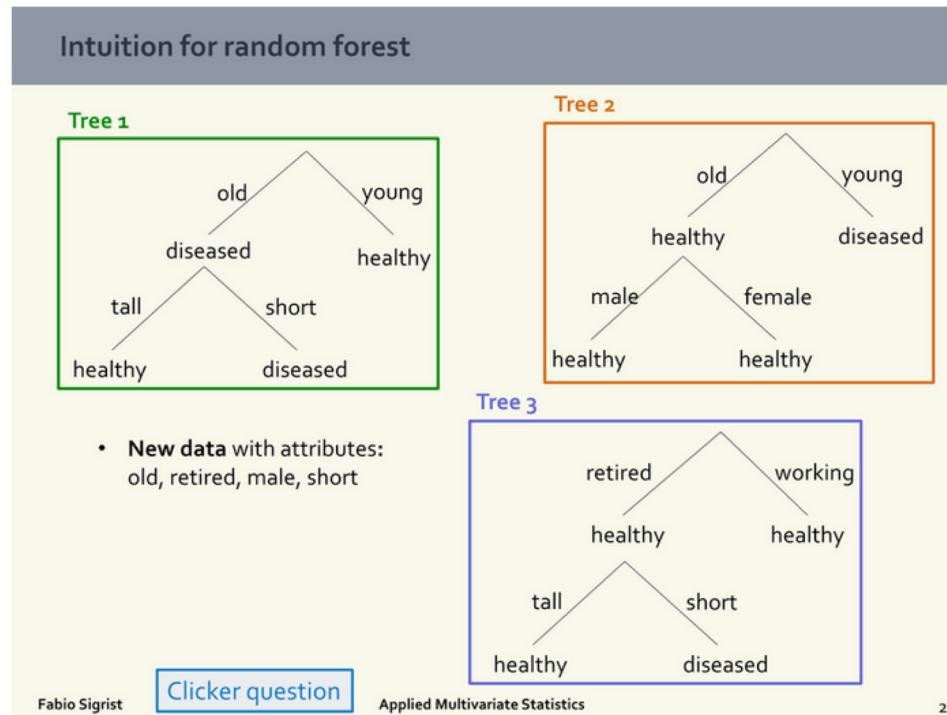
```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
max_depth=5, max_features='auto', max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=None,
oob_score=False, random_state=None, verbose=0,
warm_start=False)
```





<https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model>

5. RandomForestClassifier란.?



랜덤 포레스트의 직관에 대해서 살펴보자.

이렇게 age, working status, height, gender 총 4개의 기준 변수에 의해서 질병과 건강한 사람을 구분하는 트리를 3개 만들 수 있는데, 여기서 그 기준들을 모두 사용하지 않고 몇 개만 임의로 찍어서 사용한다. 그렇게 여러 개의 트리를 만들 수 있고, 그런 후에, 새로운 데이터를 하나 얻었다고 할 때, 그 변수가 old이고, retired이고, male이고, short이면 그 사람은 건강한가, 아니면 질병에 걸렸는가를 확인하고자 하는 것이다. Tree 1에서는 old이면서 short하면 diseased이다. Tree 2에서는 old하고, male이면 healthy하다. 그리고 Tree 3에서는 retired이면서 short하면 diseased이다. 즉, diseased가 두 개이고, healthy가 하나이다. 다수결에 의해 diseased라고 랜덤 포레스트는 예측을 하는 것이다. 이것은 너무 작은 트리들로 하여 대표적인 것의 신뢰도가 낮지만, 보다 많은 트리를 사용하면 정확률이 높아질 것이다.

추가자료 : <https://swalloow.github.io/decison-randomforest>

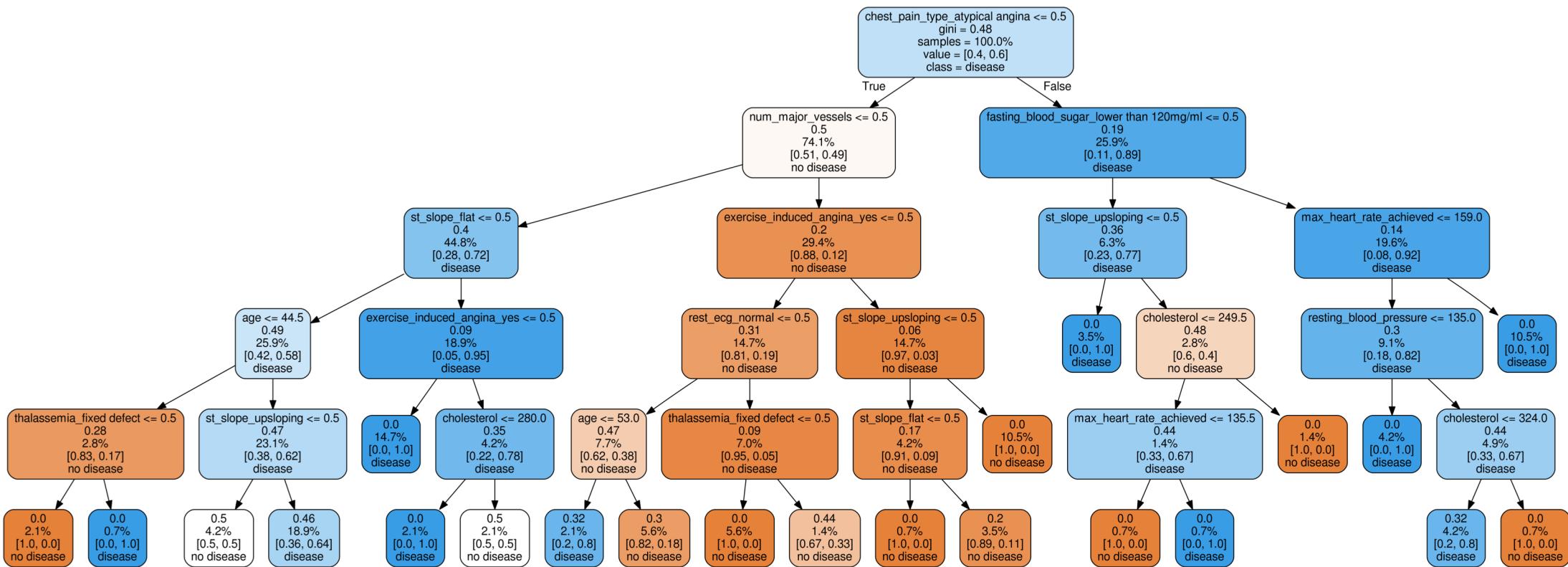


What Causes Heart Disease? Explaining the Model

Python notebook using data from [Heart Disease UCI](#) · 16,926 views · random forest, healthcare, binary classification, +1 more

<https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model>

5. RandomForestClassifier 결과





What Causes Heart Disease? Explaining the Model

Python notebook using data from [Heart Disease UCI](#) · 16,926 views · random forest, healthcare, binary classification, +1 more

<https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model>

6. 학습된 RandomForestClassifier로 결과 검증

```
y_predict = model.predict(X_test)
y_pred_quant = model.predict_proba(X_test)[:, 1]
y_pred_bin = model.predict(X_test)
```

Assess the fit with a confusion matrix,

```
confusion_matrix = confusion_matrix(y_test, y_pred_bin)
confusion_matrix
```

```
array([[28,  7],
       [ 3, 23]])
```

Diagnostic tests are often sold, marketed, cited and used with **sensitivity** and **specificity** as the headline metrics. Sensitivity and specificity are defined as,

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$

```
total=sum(sum(confusion_matrix))

sensitivity = confusion_matrix[0,0]/(confusion_matrix[0,0]+confusion_matrix[1,0])
print('Sensitivity : ', sensitivity )

specificity = confusion_matrix[1,1]/(confusion_matrix[1,1]+confusion_matrix[0,1])
print('Specificity : ', specificity)
```

```
Sensitivity :  0.9032258064516129
Specificity :  0.7666666666666667
```



What Causes Heart Disease? Explaining the Model

Python notebook using data from [Heart Disease UCI](#) · 16,926 views · random forest, healthcare, binary classification, +1 more

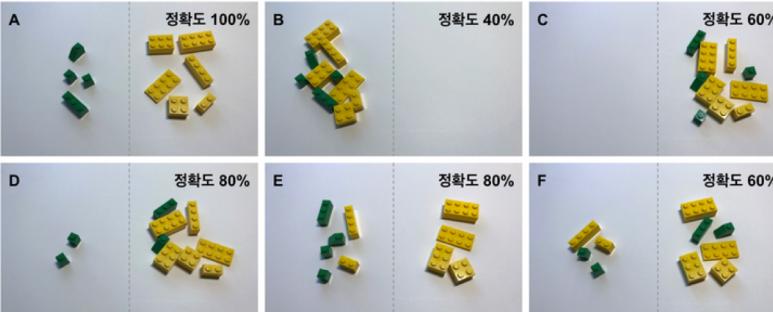
<https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model>

6. 6. 학습된 RandomForestClassifier로 결과 검증

정확도

정확도란 전체 개수 중 출수를 짹수라고 맞추고(양성을 양성이라 말하고), 짹수를 출수라고 맞춘(음성을 음성이라고 말한) 개수의 비율입니다. B모델은 전체 10개 중에 출수 블록 4개만 맞았었으므로 정확도가 40%입니다. C모델은 전체 10개 중에 짹수 블록 6개만 맞았었으므로 정확도가 60%입니다. 무조건 한 쪽으로 분류하더라도 클래스의 분포에 따라 높게 나올 수 있습니다. 만약 남자고등학교에서 남녀를 구분하는 모델을 개발한다고 했을 때, 그 모델이 무조건 남자인 결과를 내놓는다고 가정해봅시다. 실제 여자가 있다라고 모두 남자라고 분류를 하겠지만 정확도는 90%가 넘을 것입니다. 그렇다고 이 모델이 좋다고는 할 수 없습니다.

정확도를 평가하실 때는 클래스의 분포도 꼭 확인하시길 바랍니다.

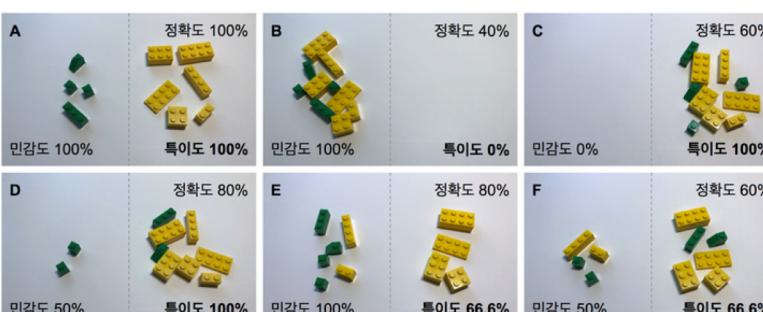


특이도

특이도는 얼마나 특이한 것만 양성으로 골라내느냐?입니다. 이말은 특이한 것만 양성으로 골라내니 반대로 음성을 음성이라고 잘 판정한다고 볼 수 있습니다.

특이도 = 판정한 것 중 실제 음성 수 / 전체 음성 수

그럼 각 모델의 특이도를 계산해보겠습니다.

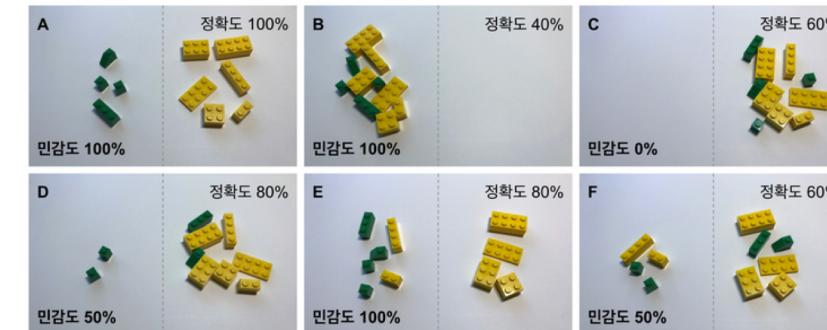


민감도

민감도는 양성에 얼마나 민감하나?라는 의미입니다. 양성을 양성이라고 판정을 잘 할수록 이 민감도가 높습니다.

민감도 = 판정한 것 중 실제 양성 수 / 전체 양성 수

그럼 각 모델에 대해서 민감도를 계산해보겠습니다.



D모델과 E모델을 다시 보겠습니다. D모델은 E모델에 비해 민감도는 낮지만 특이도는 높습니다. 만약 음성을 음성이라고 잘 골라내는 모델이 필요하다면 D모델을 선정해야 합니다. 지금까지 본 모델을 표로 비교해보겠습니다.

구분	모델 A	모델 B	모델 C	모델 D	모델 E	모델 F
맞춘 출수(전체 4개)	4개	4개	0개	2개	4개	2개
맞춘 짹수(전체 6개)	6개	0개	6개	6개	4개	4개
정확도	100%	40%	60%	80%	80%	60%
민감도	100%	100%	0%	50%	100%	50%
특이도	100%	0%	100%	100%	66.6%	66.6%

맞춘 개수는 다르지만 같은 평가 지수도 있고, 맞춘 개수는 같지만 평가 지수가 다른 것들이 보입니다. 어떤 모델이 적합한지는 문제에 따라 다르거나 꼼꼼히 생각해보도록 합시다. 도움될 만한 몇가지 예를 들어보겠습니다.

- 공항검색기는 일반물건을 위험물건이라고 잘못 판정하더라도 위험물건은 반드시 찾아야 합니다. 즉 민감도가 높아야 합니다.
- 쇼핑 시에는 꼭 필요한 물건만 구매를 해야합니다. 사야할 물건도 경우에 따라 사지 않을 수 있지만 사지 않아야 하는 물건을 반드시 안 사야합니다. 즉 특이도가 높아야 합니다.
- 자리이 나고 나면, 다음날 지진을 느낀 사람이 그럴지 않은 사람에게 있음을 겁니다. 어떤 사람(A)은 민감해서 지진도 아닌 진동도 느끼지만 웬만해도 모두 느끼는 사람이 있는 반면, 어떤 사람(B)은 정말 강도가 높은 지진이 아니고서야 웬만해서는 느끼지 못하는 사람이 있을 겁니다. 경우 다음과 같이 생각할 수 있습니다.

A가 지진을 못 느꼈다고 하면, 그날은 지진이 발생하지 않은 것입니다. 왜냐하면 A는 민감도가 높아 웬만한 지진은 다 알아내기 때문입니다.

B가 지진을 느꼈다고 하면, 그날은 지진이 발생한 것입니다. 왜냐하면 B는 특이도가 높아 지진이 발생하지 않은 것은 다 알아내기 때문입니다.

관련 자료 : https://tykimos.github.io/2017/05/22/Evaluation_Talk/



What Causes Heart Disease? Explaining the Model

Python notebook using data from [Heart Disease UCI](#) · 16,926 views · ♦ random forest, healthcare, binary classification, +1 more

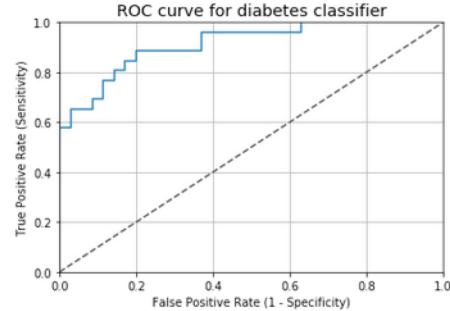
<https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model>

6. 학습된 RandomForestClassifier로 결과 검증

관련 자료 : https://tykimos.github.io/2017/05/22/Evaluation_Talk/

```
fpr, tpr, thresholds = roc_curve(y_test, y_pred_quant)

fig, ax = plt.subplots()
ax.plot(fpr, tpr)
ax.plot([0, 1], [0, 1], transform=ax.transAxes, ls="--", c=".3")
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.0])
plt.rcParams['font.size'] = 12
plt.title('ROC curve for diabetes classifier')
plt.xlabel('False Positive Rate (1 - Specificity)')
plt.ylabel('True Positive Rate (Sensitivity)')
plt.grid(True)
```



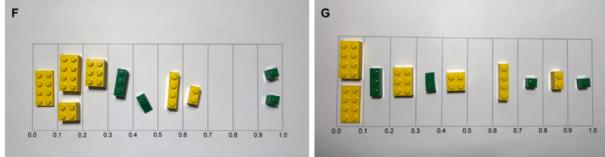
Another common metric is the **Area Under the Curve**, or AUC. This is a convenient way to capture the performance of a model in a single number, although it's not without certain issues. As a rule of thumb, an AUC can be classed as follows,

- 0.90 - 1.00 = excellent
- 0.80 - 0.90 = good
- 0.70 - 0.80 = fair
- 0.60 - 0.70 = poor
- 0.50 - 0.60 = fail

각 블록을 판정할 때는 통상적으로 모델에서는 해당 블록의 확률로 결과가 나옵니다. 즉 이 블록이 출수일 확률이 60%이거나 또는 40%이어야 이전식으로 말이죠. 이 확률로 판정결과를 나누나기 위해서 50%가 기준이 되어, 50% 이상이면 출수 블록이라고 예상하는 것 같아요. 우리는 이 50%를 임계값(threshold)이라고 부릅니다. 지금까지 위에서 봤던 결과들은 모두 확률값을 임계값을 기준으로 판정을 한 것입니다. 그럼 판정결과 이전에 확률값을 살펴보도록 하겠습니다.

F모델 결과를 보도록 하겠습니다. 총 10개 블록 중 출수 2개, 짹수 4개를 맞추었으므로 60%의 정확도를 가지고 있습니다. 민감도는 총 4개의 출수 블록 중 2개를 맞추었으니 50%입니다. 특이도는 총 6개의 짹수 블록 중 4개를 맞추었으니 66.6%입니다. 그리고 이와 동일한 정확도, 민감도, 특이도를 가진 모델G가 있다고 가정해봅시다.

F모델과 G모델이 주어진 블록에 대해 출수라고 판정한 확률값을 이 오름차순으로 나열한 뒤 10% 단위로 표시된 칸에 배치해 봤습니다.



먼저 F모델을 보겠습니다. 왼쪽의 첫번째 블록이 출수 블록일 확률이 5%라고 가정해봅시다. 그래서 0.0과 0.1 사이 칸에 위치 시킵니다. 0.5가 임계값이라고 한다면, 맞춘 출수 블록은 0.5 임계값에서 오른쪽에 있는 2개이고, 맞춘 짹수 블록은 0.5 임계값과 왼쪽에 있는 4개입니다. 이 0.5인 임계값을 조정하여 어떻게 될까요? 임계값이 0.0이라면, 모두 출수 블록이라고 하는 것으로, 맞춘 출수 블록은 4개가 되고, 맞춘 짹수 블록은 6개됩니다. 따라서, 임계값을 조정하면 정확도, 민감도, 특이도가 바뀝니다. 10% 단위로 임계값을 변화시키면서 바뀌는 정확도, 민감도, 특이도를 표정리하게 다음과 같습니다.

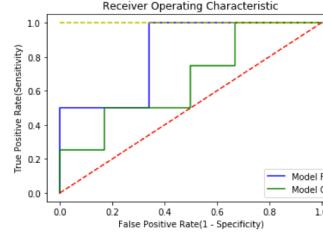
홀수 블록 임계값	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
맞춘 출수(전체4개)	4	4	4	4	3	2	2	2	2	2	0
맞춘 짹수(전체6개)	0	1	3	4	4	4	5	6	6	6	6
정확도	40%	50%	70%	80%	70%	60%	70%	80%	80%	80%	60%
민감도	100%	100%	100%	100%	75%	50%	50%	50%	50%	50%	0%
특이도	0%	16.6%	50%	66.6%	66.6%	83.3%	100%	100%	100%	100%	100%

G모델도 임계값에 따라 변화를 표로 정리해봤습니다.

홀수 블록 임계값	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
맞춘 출수(전체4개)	4	4	3	3	2	2	2	2	1	1	0
맞춘 짹수(전체6개)	0	2	2	3	3	4	4	5	5	6	6
정확도	40%	60%	50%	60%	50%	60%	60%	70%	60%	70%	60%
민감도	100%	100%	75%	75%	50%	50%	50%	25%	25%	0%	0%
특이도	0%	33.3%	33.3%	50%	50%	66.6%	66.6%	83.3%	83.3%	100%	100%

어느 모델이 더 좋을까요? 대충 보면, 모델 F가 더 좋아보입니다. 모델 F가 출수 블록이 출수일 확률이 높은 곳에 배치되어 있고, 짹수 블록이 출수일 확률이 낮은 곳에 배치되어 있기 때문입니다. 이런 패턴을 보면서 많이 사용되는 것이 ROC(Receiver Operating Characteristic) curve입니다. 이는 민감도와 특이도가 어떤 관계를 가지고 변하는지 그려놓은 것입니다. 이러한 ROC curve 아래 면적을 구한 값을 AUC(Area Under Curve)이라고 하는데, 하나의 수치로 계산되어서 성능 비교를 간단히 할 수 있습니다.

ROC curve를 그리는 방법은 간단합니다. 각 임계값으로 민감도와 특이도를 계산하여 x축을 (1-특이도), y축을 민감도로 두어서 이차원 평면 상에 점을 찍고 연결하면 됩니다. 모델 F와 모델 G 대해서 ROC Curve를 그리는 소스코드와 결과는 다음과 같습니다.



마지막으로 노란점선이 이상적인 모델을 표시한 것입니다. 임계값과 상관없이 민감도와 특이도가 100%일때를 말하고, AUC 값은 1입니다. 빨간점선은 '기준선'으로서 AUC 값이 0.5입니다. 개발한 모델을 사용하면서 적어도 이 기준선보다는 상위에 있어야 되겠죠? 모델 F와 모델 G를 비교해보면, 모델 F의 AUC가 모델 G보다 상위에 있음을 알 수 있습니다. AUC를 보더라도 모델 F가 훨씬 더 낫습니다. sklearn 패키지는 ROC curve 및 AUC를 좀 더 쉽게 구할 수 있는 함수를 제공합니다. 임계값 변화에 따른 민감도, 특이도를 계산해서 입력할 필요없이, 클래스 갯수와 모델에서 나오는 클래스 확률값을 그대로 입력하면, ROC curve를 그릴 수 있는 값과 AUC 값을 알려줍니다. sklearn 패키지를 이용한 소스코드는 다음과 같습니다.



What Causes Heart Disease? Explaining the Model

Python notebook using data from [Heart Disease UCI](#) · 16,926 views · random forest, healthcare, binary classification, +1 more

<https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model>

7. 학습된 RandomForestClassifier 결과에 영향을 미치는 주요 인자 확인

```
perm = PermutationImportance(model, random_state=1).fit(X_test, y_test)
eli5.show_weights(perm, feature_names = X_test.columns.tolist())
```

Weight	Feature
0.0754 ± 0.0445	thalassemia_reversible defect
0.0459 ± 0.0321	max_heart_rate_achieved
0.0393 ± 0.0334	num_major_vessels
0.0295 ± 0.0245	st_depression
0.0197 ± 0.0382	thalassemia_fixed defect
0.0197 ± 0.0245	rest_ecg_normal
0.0164 ± 0.0000	exercise_induced_angina_yes
0.0131 ± 0.0131	sex_male
0.0131 ± 0.0131	cholesterol
0.0066 ± 0.0334	age
0 ± 0.0000	chest_pain_type_non-anginal pain
0 ± 0.0000	thalassemia_normal
0 ± 0.0000	fasting_blood_sugar_lower than 120mg/ml
0 ± 0.0000	rest_ecg_left ventricular hypertrophy
0 ± 0.0000	st_slope_upsloping
0 ± 0.0000	resting_blood_pressure
-0.0033 ± 0.0131	chest_pain_type_atypical angina
-0.0131 ± 0.0131	chest_pain_type_typical angina
-0.0328 ± 0.0207	st_slope_flat

Permutation Importance : https://eli5.readthedocs.io/en/latest/blackbox/permuation_importance.html
Kaggle example : <https://www.kaggle.com/dansbecker/permuation-importance>

So, it looks like the most important factors in terms of permutation is a thalassemia result of 'reversible defect'. The high importance of 'max heart rate achieved' type makes sense, as this is the immediate, subjective state of the patient at the time of examination (as opposed to, say, age, which is a much more general factor).

Let's take a closer look at the number of major vessels using a **Partial Dependence Plot** (learn more [here](#)). These plots vary a single variable in a single row across a range of values and see what effect it has on the outcome. It does this for several rows and plots the average effect. Let's take a look at the 'num_major_vessels' variable, which was at the top of the permutation importance list,



What Causes Heart Disease? Explaining the Model

Python notebook using data from [Heart Disease UCI](#) · 16,926 views · random forest, healthcare, binary classification, +1 more

<https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model>

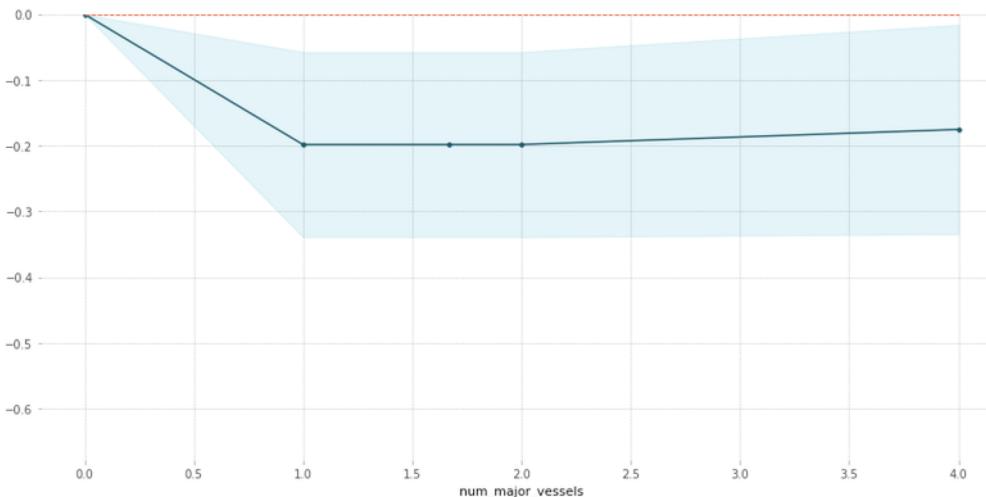
7. 학습된 RandomForestClassifier 결과에 영향을 미치는 주요 인자 확인(PDP)

```
base_features = dt.columns.values.tolist()
base_features.remove('target')

feat_name = 'num_major_vessels'
pdp_dist = pdp.pdp_isolate(model=model, dataset=X_test, model_features=base_features, feature=feat_name)

pdp.pdp_plot(pdp_dist, feat_name)
plt.show()
```

PPD for feature "num_major_vessels"
Number of unique grid points: 5



Partial Dependence Plots : https://scikit-learn.org/stable/auto_examples/ensemble/plot_partial_dependence.html
Kaggle example : <https://christophm.github.io/interpretable-ml-book/pdp.html>

심장으로 가는 주요 혈관이 많을 수록, 심장병 질환 발생 가능성이 낮아짐

So, we can see that as the number of major blood vessels *increases*, the probability of heart disease *decreases*. That makes sense, as it means more blood can get to the heart.



What Causes Heart Disease? Explaining the Model

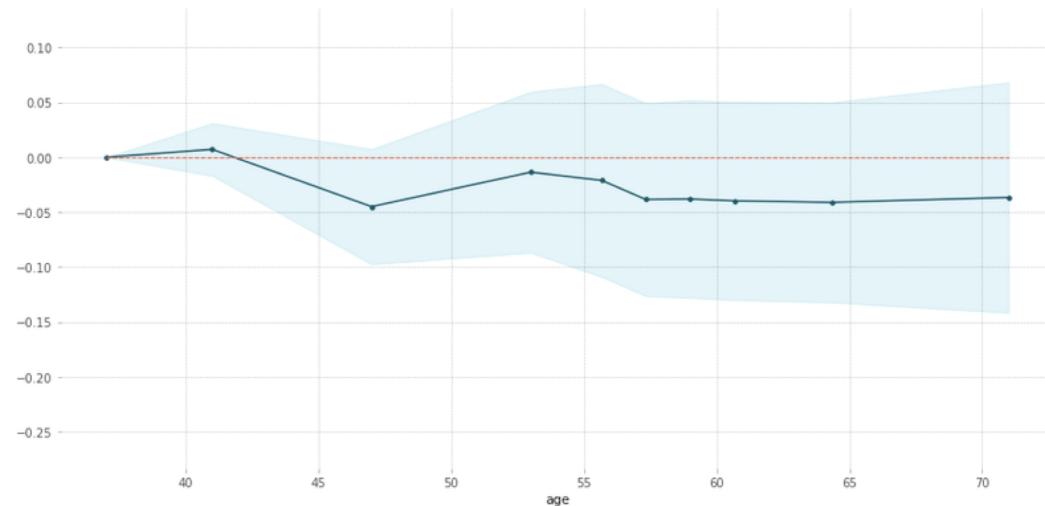
Python notebook using data from [Heart Disease UCI](#) · 16,926 views · random forest, healthcare, binary classification, +1 more

<https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model>

7. 학습된 RandomForestClassifier 결과에 영향을 미치는 주요 인자 확인(PDP)

```
feat_name = 'age'  
pdp_dist = pdp.pdp_isolate(model=model, dataset=X_test, model_features=base_features, feature=feat_name)  
  
pdp.pdp_plot(pdp_dist, feat_name)  
plt.show()
```

PDP for feature "age"
Number of unique grid points: 10



Partial Dependence Plots : https://scikit-learn.org/stable/auto_examples/ensemble/plot_partial_dependence.html
Kaggle example : <https://christophm.github.io/interpretable-ml-book/pdp.html>

나이가 많을 수록, 심장병 질환 발생 가능성이 높아 지지는 않음.

That's a bit odd. The higher the age, the lower the chance of heart disease? Although the blue confidence regions show that this might not be true (the red baseline is within the blue zone).



What Causes Heart Disease? Explaining the Model

Python notebook using data from [Heart Disease UCI](#) · 16,926 views · random forest, healthcare, binary classification, +1 more

<https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model>

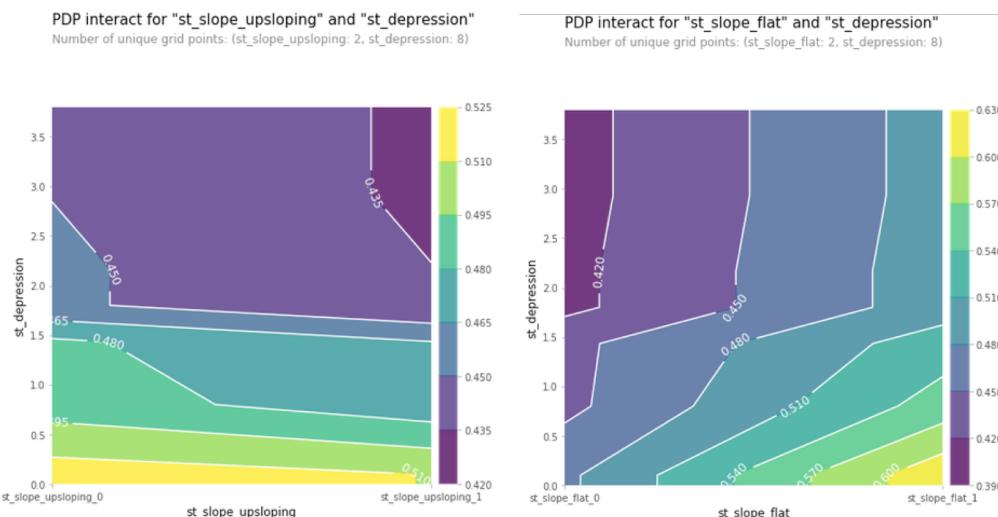
7. 학습된 RandomForestClassifier 결과에 영향을 미치는 주요 인자 확인(Contour Plot)

```
inter1 = pdp.pdp_interact(model=model, dataset=X_test, model_features=base_features, features=['st_slope_upsloping', 'st_depression'])

pdp.pdp_interact_plot(pdp_interact_out=inter1, feature_names=['st_slope_upsloping', 'st_depression'], plot_type='contour')
plt.show()

inter1 = pdp.pdp_interact(model=model, dataset=X_test, model_features=base_features, features=['st_slope_flat', 'st_depression'])

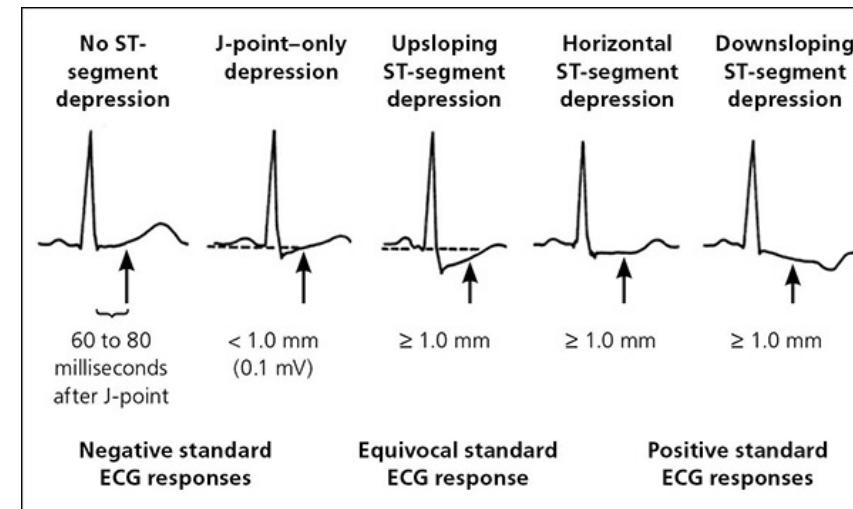
pdp.pdp_interact_plot(pdp_interact_out=inter1, feature_names=['st_slope_flat', 'st_depression'], plot_type='contour')
plt.show()
```



Contour Plot : https://matplotlib.org/api/_as_gen/matplotlib.pyplot.contour.html

Contour Plot : https://matplotlib.org/gallery/images_contours_and_fields/contour_demo.html

st_slope_flat, st_slope_upsloping와 심장질환과의 관계는 연관성이 낮음.



운동 할 때의 심전도 값



What Causes Heart Disease? Explaining the Model

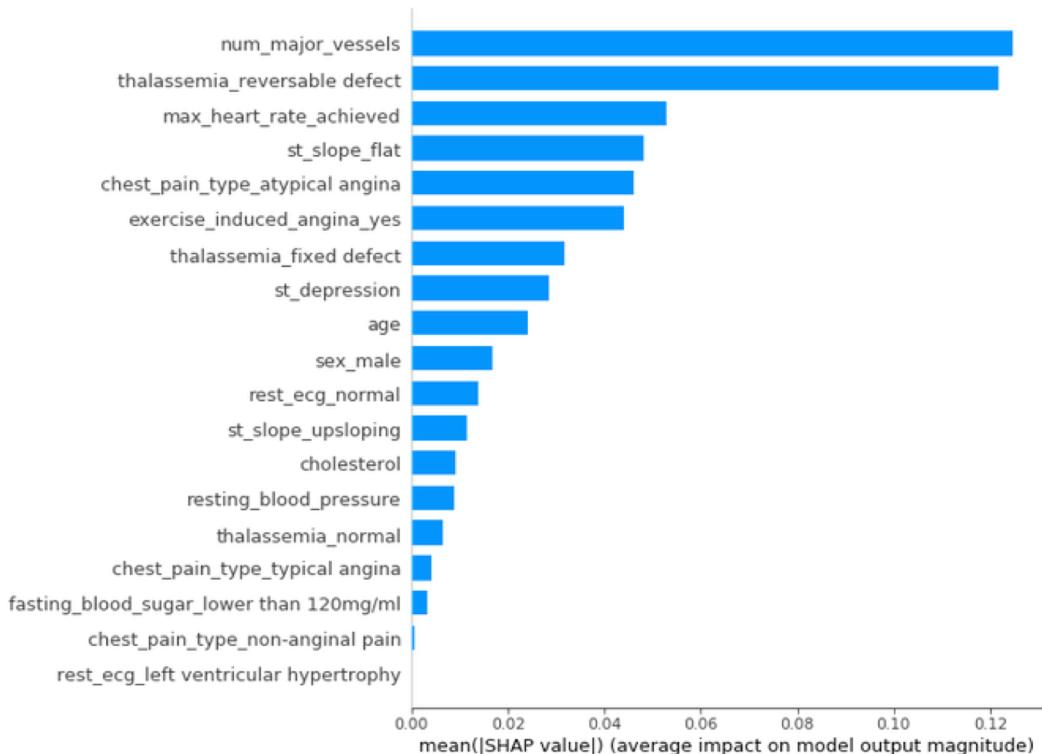
Python notebook using data from [Heart Disease UCI](#) · 16,926 views · random forest, healthcare, binary classification, +1 more

<https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model>

7. 학습된 RandomForestClassifier 결과에 영향을 미치는 주요 인자 확인 (SHAP)

```
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X_test)

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



SHAP Values : <https://brunch.co.kr/@bdh/26>

Contour Plot : https://matplotlib.org/gallery/images_contours_and_fields/contour_demo.html

다음으로 SHAP에 대해 알아보겠습니다.

SHAP는 Shapley Additive exPlanations의 약자로 이름에서 알 수 있듯 Shapley 값과 관련이 있습니다.

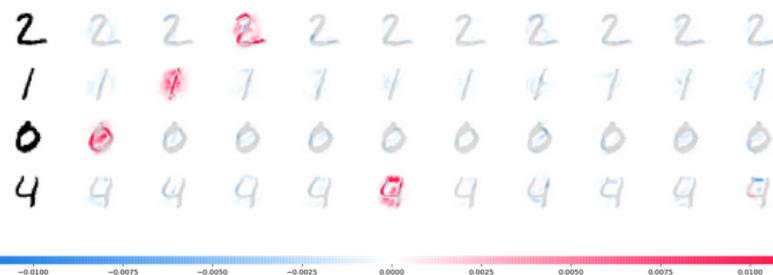
이 Shapley 값은 특정 결과에 각 공헌자가 얼마나 공헌했는지를 나타내는 수치로 게임 이론에 나오는 개념이라고 하네요. [위키피디아](#)에서 가져온 정의 중 수식이 아닌 설명으로 된 부분을 가져왔습니다.

$$\phi_i(v) = \frac{1}{\text{number of players}} \sum_{\text{coalitions excluding } i} \frac{\text{marginal contribution of } i \text{ to coalition}}{\text{number of coalitions excluding } i \text{ of this size}}$$

직관적으로 설명하자면 [(A라는 플레이어의 행위를 포함한 결과)에서 (marginal하게 A 플레이어의 행위를 제외한 결과)를 뺀 값]입니다. 이 값이 작으면 A 플레이어의 플레이를 무시했는데도 포함됐을 때랑 결과에 큰 차이가 없다는 것으로 A의 공헌도는 낮습니다. 반대로 이 값이 크면 A 플레이어의 플레이를 무시했더니 포함했을 때랑 차이가 커졌으므로 A의 공헌도는 높습니다.

이런 Shapley 기댓값 개념과 몇몇 머신러닝 해석 개념을 활용해 구현한 것이 SHAP라고 볼 수 있으며, 모델 타입마다 다른 SHAP 산식을 사용할 수 있어 방법을 딱 잘라 얘기할 수 없지만 기존 Shapley 값을 구하는 방식보다 쉽게 구할 수 있습니다.

개인적으로는 아래 그림을 보고 직관적으로 확 와 달았습니다. 2,1,0,4의 값을 넣었을 때 그것을 각각 2,1,0,4로 분류하게 한 과거 데이터 중 10개의 공헌도를 표현한 것입니다. 2를 나타내는 여려 그림 중 3번째 그림의 많은 부분이 2라고 해석하는 SHAP 값이 높았습니다. 4를 분류할 때 중요한 것은 9와 잘 구별하는 것인데 4 그림의 윗부분이 빨간 것을 보면 4의 윗부분이 9와 구분해주는 중요한 차이라는 것을 알 수 있습니다.



출처: <https://github.com/slundberg/shap>

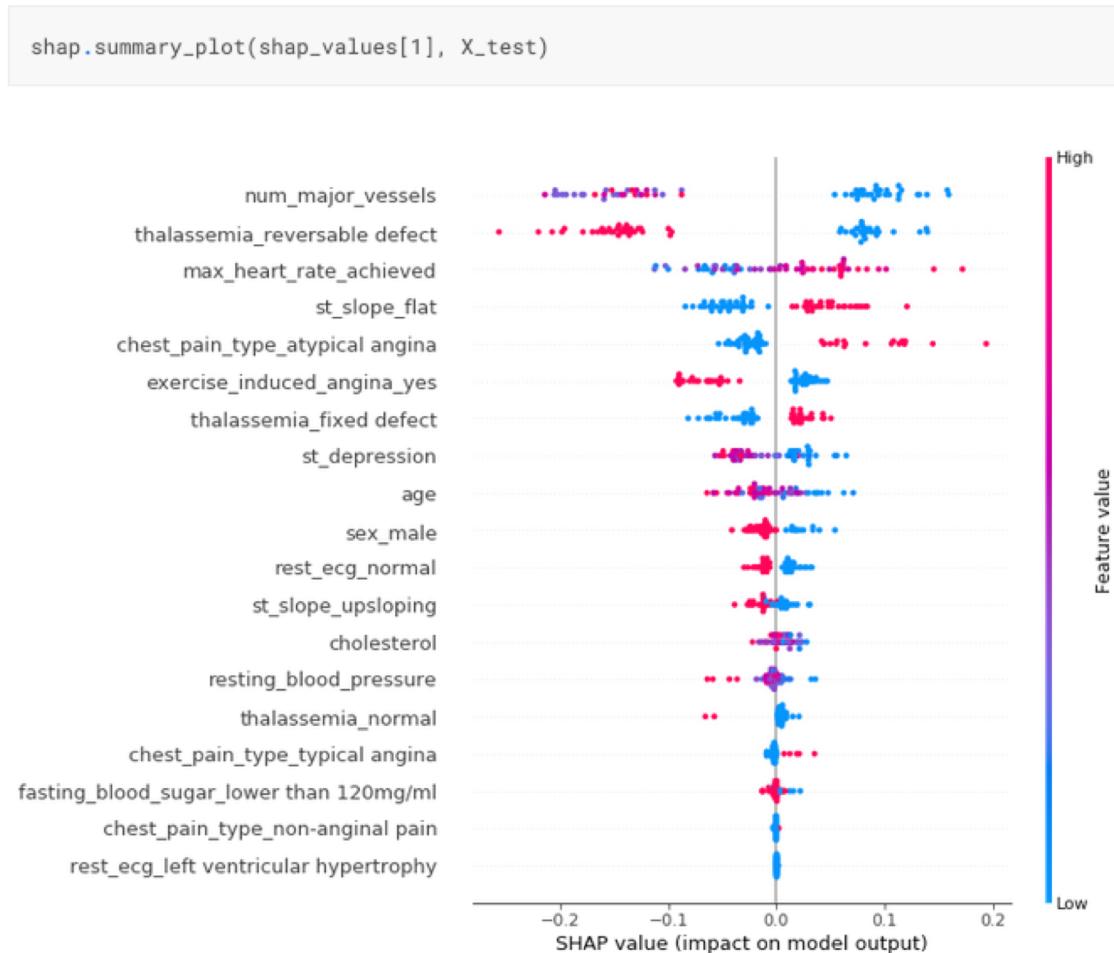


What Causes Heart Disease? Explaining the Model

Python notebook using data from [Heart Disease UCI](#) · 16,926 views · random forest, healthcare, binary classification, +1 more

<https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model>

7. 학습된 RandomForestClassifier 결과에 영향을 미치는 주요 인자 확인 (SHAP)



SHAP Values : <https://brunch.co.kr/@bdh/26>

다음으로 SHAP에 대해 알아보겠습니다.

SHAP는 Shapley Additive exPlanations의 약자로 이름에서 알 수 있듯 Shapley 값과 관련이 있습니다.

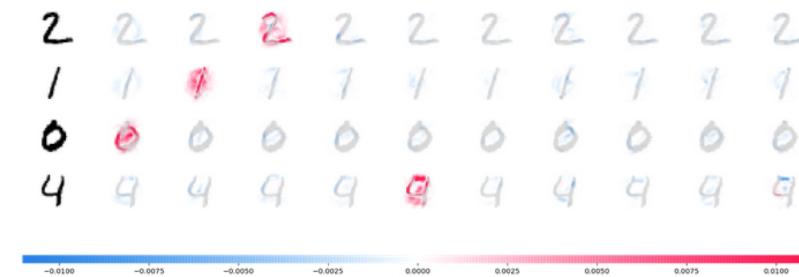
이 Shapley 값은 특정 결과에 각 공헌자가 얼마나 공헌했는지를 나타내는 수치로 게임 이론에 나오는 개념이라고 하네요. [위키피디아](#)에서 가져온 정의 중 수식이 아닌 설명으로 된 부분을 가져왔습니다.

$$\phi_i(v) = \frac{1}{\text{number of players}} \sum_{\text{coalitions excluding } i} \frac{\text{marginal contribution of } i \text{ to coalition}}{\text{number of coalitions excluding } i \text{ of this size}}$$

직관적으로 설명하자면 [(A라는 플레이어의 행위를 포함한 결과)에서 (marginal하게 A 플레이어의 행위를 제외한 결과)를 뺀 값]입니다. 이 값이 작으면 A 플레이어의 플레이를 무시했는데도 포함됐을 때랑 결과에 큰 차이가 없다는 것으로 A의 공헌도는 낮습니다. 반대로 이 값이 크면 A 플레이어의 플레이를 무시했더니 포함했을 때랑 차이가 커졌으므로 A의 공헌도는 높습니다.

이런 Shapley 기댓값 개념과 몇몇 머신러닝 해석 개념을 활용해 구현한 것이 SHAP라고 볼 수 있으며, 모델 타입마다 다른 SHAP 산식을 사용할 수 있어 방법을 딱 잘라 얘기할 수 없지만 기존 Shapley 값을 구하는 방식보다 쉽게 구할 수 있습니다.

개인적으로는 아래 그림을 보고 직관적으로 확 와 달았습니다. 2,1,0,4의 값을 넣었을 때 그것을 각각 2,1,0,4로 분류하게 한 과거 데이터 중 10개의 공헌도를 표현한 것입니다. 2를 나타내는 여려 그림 중 3번째 그림의 많은 부분이 2라고 해석하는 SHAP 값이 높았습니다. 4를 분류할 때 중요한 것은 9와 잘 구별하는 것인데 4 그림의 윗부분이 빨간 것을 보면 4의 윗부분이 9와 구분해주는 중요한 차이라는 것을 알 수 있습니다.



출처: <https://github.com/slundberg/shap>



What Causes Heart Disease? Explaining the Model

Python notebook using data from [Heart Disease UCI](#) · 16,926 views · random forest, healthcare, binary classification, +1 more

<https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model>

7. 학습된 RandomForestClassifier 결과에 영향을 미치는 주요 인자 확인 (SHAP)

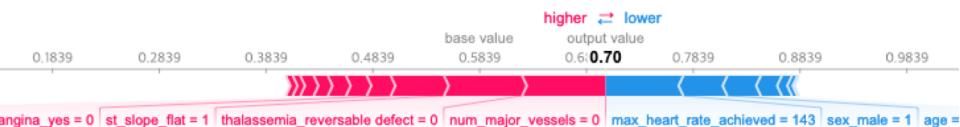
```
def heart_disease_risk_factors(model, patient):

    explainer = shap.TreeExplainer(model)
    shap_values = explainer.shap_values(patient)
    shap.initjs()
    return shap.force_plot(explainer.expected_value[1], shap_values[1], patient)
```

```
data_for_prediction = X_test.iloc[1,:].astype(float)
heart_disease_risk_factors(model, data_for_prediction)
```

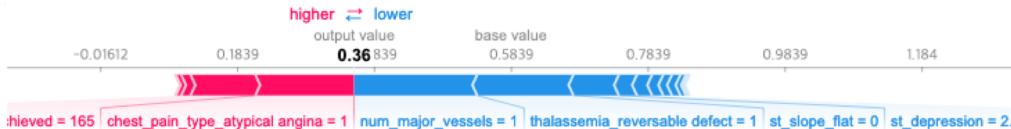
```
data_for_prediction = X_test.iloc[3,:].astype(float)
heart_disease_risk_factors(model, data_for_prediction)
```

js

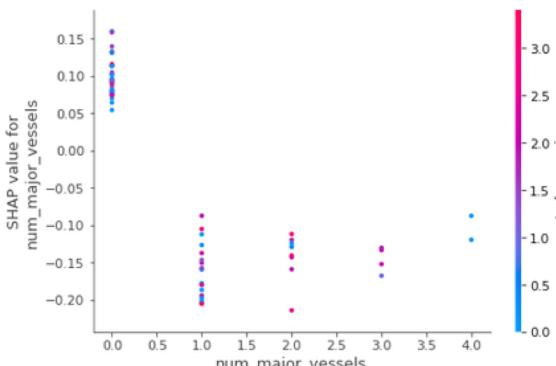


For this person, their prediction is 70% (compared to a baseline of 58.4%). Not working in their favour are things like having no major vessels, a flat st_slope, and *not* a reversible thalassemia defect.

We can also plot something called 'SHAP dependence contribution plots' (learn more [here](#)), which are pretty self-explanatory in the context of SHAP values,



```
ax2 = fig.add_subplot(224)
shap.dependence_plot('num_major_vessels', shap_values[1], X_test, interaction_index="st_depression")
```





What Causes Heart Disease? Explaining the Model

Python notebook using data from [Heart Disease UCI](#) · 16,926 views · random forest, healthcare, binary classification, +1 more

<https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model>

7. 학습된 RandomForestClassifier 결과에 영향을 미치는 주요 인자 확인 (SHAP)

The final plot, for me, is one of the most effective. It shows the predictions and influencing factors for many (in this case 50) patients, all together. It's also interactive, which is great. Hover over to see *why* each person ended up either red (prediction of disease) or blue (prediction of no disease),

```
shap_values = explainer.shap_values(X_train.iloc[:50])
shap.force_plot(explainer.expected_value[1], shap_values[1], X_test.iloc[:50])
```

