

```
In [1]: import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: df=pd.read_csv('heart.csv')
```

```
In [15]: !pip uninstall -y scikit-learn
!pip install scikit-learn==1.3.1
```

Found existing installation: scikit-learn 1.6.1

Uninstalling scikit-learn-1.6.1:

Successfully uninstalled scikit-learn-1.6.1

WARNING: Failed to remove contents in a temporary directory 'C:\Users\user\AppData\Roaming\Python\Python312\site-packages\~learn'.

You can safely remove it manually.

```
Defaulting to user installation because normal site-packages is not writeable
Collecting scikit-learn==1.3.1
    Downloading scikit_learn-1.3.1-cp312-cp312-win_amd64.whl.metadata (11 kB)
Requirement already satisfied: numpy<2.0,>=1.17.3 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn==1.3.1) (1.26.4)
Requirement already satisfied: scipy>=1.5.0 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn==1.3.1) (1.13.1)
Requirement already satisfied: joblib>=1.1.1 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn==1.3.1) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\user\appdata\roaming\python\python312\site-packages (from scikit-learn==1.3.1) (3.5.0)
Downloading scikit_learn-1.3.1-cp312-cp312-win_amd64.whl (9.1 MB)
----- 0.0/9.1 MB ? eta ------
----- 0.0/9.1 MB ? eta ------
----- 0.2/9.1 MB 2.8 MB/s eta 0:00:04
----- 0.7/9.1 MB 5.3 MB/s eta 0:00:02
----- 0.9/9.1 MB 5.8 MB/s eta 0:00:02
----- 0.9/9.1 MB 5.8 MB/s eta 0:00:02
----- 0.9/9.1 MB 5.8 MB/s eta 0:00:02
----- 0.9/9.1 MB 3.5 MB/s eta 0:00:03
----- 1.0/9.1 MB 2.8 MB/s eta 0:00:03
----- 1.1/9.1 MB 1.8 MB/s eta 0:00:05
----- 1.6/9.1 MB 2.5 MB/s eta 0:00:03
----- 2.1/9.1 MB 3.1 MB/s eta 0:00:03
----- 2.7/9.1 MB 3.0 MB/s eta 0:00:03
----- 3.1/9.1 MB 3.4 MB/s eta 0:00:02
----- 3.6/9.1 MB 2.7 MB/s eta 0:00:03
----- 4.2/9.1 MB 3.0 MB/s eta 0:00:02
```

```
----- 4.2/9.1 MB 3.0 MB/s eta 0:00:02
----- 4.2/9.1 MB 2.0 MB/s eta 0:00:03
----- 4.8/9.1 MB 2.2 MB/s eta 0:00:02
----- 5.2/9.1 MB 2.4 MB/s eta 0:00:02
----- 5.7/9.1 MB 2.3 MB/s eta 0:00:02
----- 6.3/9.1 MB 2.5 MB/s eta 0:00:02
----- 6.4/9.1 MB 2.3 MB/s eta 0:00:02
----- 6.9/9.1 MB 2.4 MB/s eta 0:00:01
----- 7.3/9.1 MB 2.6 MB/s eta 0:00:01
----- 7.4/9.1 MB 2.4 MB/s eta 0:00:01
----- 7.9/9.1 MB 2.5 MB/s eta 0:00:01
----- 8.4/9.1 MB 2.6 MB/s eta 0:00:01
----- 8.7/9.1 MB 2.5 MB/s eta 0:00:01
----- 9.1/9.1 MB 2.6 MB/s eta 0:00:01
----- 9.1/9.1 MB 2.6 MB/s eta 0:00:01
----- 9.1/9.1 MB 2.6 MB/s eta 0:00:00
```

Installing collected packages: scikit-learn

Successfully installed scikit-learn-1.3.1

ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.

category-encoders 2.8.0 requires scikit-learn>=1.6.0, but you have scikit-learn 1.3.1 which is incompatible.

In [3]: df

Out[3]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3

1025 rows × 14 columns



In [4]: `pip install --upgrade xgboost scikit-learn`

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: xgboost in c:\users\user\appdata\roaming\python\python312\site-packages (2.1.3)
Requirement already satisfied: scikit-learn in c:\users\user\appdata\roaming\python\python312\site-packages (1.5.2)
Collecting scikit-learn
  Using cached scikit_learn-1.6.1-cp312-cp312-win_amd64.whl.metadata (15 kB)
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-packages (from xgboost) (1.26.4)
Requirement already satisfied: scipy in c:\programdata\anaconda3\lib\site-packages (from xgboost) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\user\appdata\roaming\python\python312\site-packages (from scikit-learn) (3.5.0)
Using cached scikit_learn-1.6.1-cp312-cp312-win_amd64.whl (11.1 MB)
Installing collected packages: scikit-learn
  Attempting uninstall: scikit-learn
    Found existing installation: scikit-learn 1.5.2
    Uninstalling scikit-learn-1.5.2:
      Successfully uninstalled scikit-learn-1.5.2
Successfully installed scikit-learn-1.6.1
Note: you may need to restart the kernel to use updated packages.
```

In [6]: `pip install hyperopt`

```

Defaulting to user installation because normal site-packages is not writeable
Collecting hyperopt
  Downloading hyperopt-0.2.7-py2.py3-none-any.whl.metadata (1.7 kB)
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-packages
  (from hyperopt) (1.26.4)
Requirement already satisfied: scipy in c:\programdata\anaconda3\lib\site-packages
  (from hyperopt) (1.13.1)
Requirement already satisfied: six in c:\programdata\anaconda3\lib\site-packages
  (from hyperopt) (1.16.0)
Requirement already satisfied: networkx>=2.2 in c:\programdata\anaconda3\lib\site-packages
  (from hyperopt) (3.2.1)
Collecting future (from hyperopt)
  Downloading future-1.0.0-py3-none-any.whl.metadata (4.0 kB)
Requirement already satisfied: tqdm in c:\programdata\anaconda3\lib\site-packages
  (from hyperopt) (4.66.4)
Requirement already satisfied: cloudpickle in c:\programdata\anaconda3\lib\site-packages
  (from hyperopt) (2.2.1)
Collecting py4j (from hyperopt)
  Downloading py4j-0.10.9.9-py2.py3-none-any.whl.metadata (1.3 kB)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-packages
  (from tqdm->hyperopt) (0.4.6)
Downloading hyperopt-0.2.7-py2.py3-none-any.whl (1.6 MB)
----- 0.0/1.6 MB ? eta -:---:
----- 0.0/1.6 MB 682.7 kB/s eta 0:00:03
----- 0.1/1.6 MB 1.3 MB/s eta 0:00:02
----- 0.4/1.6 MB 3.9 MB/s eta 0:00:01
----- 0.8/1.6 MB 5.5 MB/s eta 0:00:01
----- 1.2/1.6 MB 6.5 MB/s eta 0:00:01
----- 1.6/1.6 MB 6.7 MB/s eta 0:00:01
----- 1.6/1.6 MB 5.9 MB/s eta 0:00:00
Downloading future-1.0.0-py3-none-any.whl (491 kB)
----- 0.0/491.3 kB ? eta -:---:
----- 399.4/491.3 kB 12.6 MB/s eta 0:00:01
----- 491.3/491.3 kB 7.8 MB/s eta 0:00:00
Downloading py4j-0.10.9.9-py2.py3-none-any.whl (203 kB)
----- 0.0/203.0 kB ? eta -:---:
----- 203.0/203.0 kB 6.2 MB/s eta 0:00:00
Installing collected packages: py4j, future, hyperopt
Successfully installed future-1.0.0 hyperopt-0.2.7 py4j-0.10.9.9
Note: you may need to restart the kernel to use updated packages.

```

WARNING: The scripts futurize.exe and pasteurize.exe are installed in 'C:\Users\user\AppData\Roaming\Python\Python312\Scripts' which is not on PATH.

Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.

WARNING: The script hyperopt-mongo-worker.exe is installed in 'C:\Users\user\AppData\Roaming\Python\Python312\Scripts' which is not on PATH.

Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.

In [4]:

```

import xgboost as xgb
from sklearn.metrics import accuracy_score

```

```

# import packages for hyperparameters tuning
from hyperopt import STATUS_OK, Trials, fmin, hp, tpe

```

```
In [5]: X = df.drop('target', axis=1)

y = df['target']
```

```
In [6]: from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_s
```

```
In [7]: space={'max_depth': hp.quniform("max_depth", 3, 18, 1),
           'gamma': hp.uniform ('gamma', 1,9),
           'reg_alpha' : hp.quniform('reg_alpha', 0,180,1),
           'reg_lambda' : hp.uniform('reg_lambda', 0,1),
           'colsample_bytree' : hp.uniform('colsample_bytree', 0.5,1),
           'min_child_weight' : hp.quniform('min_child_weight', 0, 10, 1),
           'n_estimators': hp.choice('n_estimators', [100, 200, 400, 600]),
           'seed': 0
      }
```

```
In [8]: from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

```
In [9]: def objective(space):
    clf=xgb.XGBClassifier(
        n_estimators =space['n_estimators'], max_depth = int(space['max_depth']),
        reg_alpha = int(space['reg_alpha']),min_child_weight=int(space['min_child_weight']),
        colsample_bytree=int(space['colsample_bytree']),enable_categorical=True)

    evaluation = [( X_train, y_train), ( X_test, y_test)]

    clf.fit(X_train, y_train,
            eval_set=evaluation,
            verbose=False)

    pred = clf.predict(X_test)
    accuracy = accuracy_score(y_test, pred>0.5)
    print ("SCORE:", accuracy)
    return {'loss': -accuracy, 'status': STATUS_OK }
```

```
In [10]: trials = Trials()

best_hyperparams = fmin(fn = objective,
                       space = space,
                       algo = tpe.suggest,
                       max_evals = 10,
                       trials = trials)
```

```

SCORE:
0.8603896103896104
SCORE:
0.8603896103896104
SCORE:
0.7792207792207793
SCORE:
0.5292207792207793
SCORE:
0.8701298701298701
SCORE:
0.5292207792207793
SCORE:
0.5292207792207793
SCORE:
0.8603896103896104
SCORE:
0.5292207792207793
SCORE:
0.8636363636363636
100%|██████████| 10/10 [00:02<00:00,  3.47tria
1/s, best loss: -0.8701298701298701]

```

In [11]: `print("The best hyperparameters are : ", "\n")  
print(best_hyperparams)`

The best hyperparameters are :

```
{'colsample_bytree': 0.7198606641426617, 'gamma': 3.8786597911057497, 'max_depth': 1
1.0, 'min_child_weight': 1.0, 'n_estimators': 1, 'reg_alpha': 42.0, 'reg_lambda': 0.
4336645171006904}
```

In [12]: `params={'colsample_bytree': 0.7952000816411979, 'gamma': 1.9113124584358845, 'max_d`

In [13]: `best_model = xgb.XGBClassifier(**params, use_label_encoder=False, eval_metric='mlog
best_model.fit(X_train, y_train)`

Out[13]: ▾ XGBClassifier

```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=0.7952000816411979, device=None,
              early_stopping_rounds=None, enable_categorical=False,
              eval_metric='mlogloss', feature_types=None,
              gamma=1.9113124584358845, grow_policy=None, importance_ty
              pe=None,
              interaction_constraints=None, learning_rate=None, max_bin
              =None,
```

In [15]: `best_model.fit(X_train, y_train,
verbose=False)`

Out[15]:

```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=0.7952000816411979, device=None,
              early_stopping_rounds=None, enable_categorical=False,
              eval_metric='mlogloss', feature_types=None,
              gamma=1.9113124584358845, grow_policy=None, importance_ty
              pe=None,
              interaction_constraints=None, learning_rate=None, max_bin
              =None,
```

In [17]:

```
y_pred=best_model.predict(X_test)
```

In [18]:

```
accuracy = accuracy_score(y_test, y_pred)
print(f"AdaBoost Accuracy on the test Dataset: {accuracy:.2f}")
```

AdaBoost Accuracy on the test Dataset: 0.77

In [19]:

```
# Print the Confusion Matrix and slice it into four pieces

from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)

print('Confusion matrix\n\n', cm)
```

Confusion matrix

```
[[ 77  68]
 [  3 160]]
```

In [20]:

```
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.96	0.53	0.68	145
1	0.70	0.98	0.82	163
accuracy			0.77	308
macro avg	0.83	0.76	0.75	308
weighted avg	0.82	0.77	0.76	308

In [21]:

```
y_pred_proba=best_model.predict_proba(X_test)
```

In [22]:

```
from sklearn.metrics import roc_curve
fpr, tpr, thresholds = roc_curve(y_test, y_pred_proba[:,1])
plt.figure(figsize=(6,4))
plt.plot(fpr, tpr, linewidth=2)
plt.plot([0,1], [0,1], 'k--')
```

```
plt.title('ROC curve for classifier')
plt.xlabel('False Positive Rate (1 - Specificity)')
plt.ylabel('True Positive Rate (Sensitivity)')
plt.show()
```

