FINAL PROJECT:

DEEP LEARNING AND REINFORCEMENT LEARNING

Antonio Jimenez

IBM ON COURSERA . March, 2023

Table of Content

Introduction

The ministry of heath of the Plurinational State of Bolivia is creating a public insurance program that will cover heart related diseases. In order to evaluate the budget needed for this program, they have collected a data set that will be used to create a model that will predict whether or not a person has high chances of developing heart related disease. This model will then be used with the data that is going to be collected from the people at the moment of registration to determine the amount of resources needed for this program to run. The model has to have high prediction power but, in addition, interpretability is required.

Description of the Data

The data chosen was downloaded from Kaggle under the name of "Heart Attack Analysis & Prediction Dataset" and uploaded by Rashik Rahman, with 303 observations. The original dataset was collected by:

1. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.
2. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.
3. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.
4. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.

The features of the dataset are the following (Rashik):

Age: Age of the patient

Sex: Sex of the patient. (0 = female, 1 = male)

cp: Chest Pain type. Value 0: typical angina, Value 1: atypical angina, Value 2: non-anginal pain, Value 3: asymptomatic.

trtbps: resting blood pressure (in mm Hg)

chol: cholesterol in mg/dl fetched via BMI sensor

fbs: (fasting blood sugar > 120 mg/dl). 1 = true, 0 = false.

rest_ecg: resting electrocardiographic results. Value 0: normal, Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria.

thalach: maximum heart rate achieved

exang: exercise induced angina. 1 = yes, 0 = no

old peak: ST depression induced by exercise relative to rest

slp: the slope of the peak exercise ST segment. 0 = unsloping, 1 = flat, 2 = downsloping.

caa: number of major vessels (0-3)

thall: thalassemia. 0 = null, 1 = fixed defect, 2 = normal, 3 = reversable defect.

output: diagnosis of heart disease (angiographic disease status)

0: < 50% diameter narrowing. less chance of heart disease

1: > 50% diameter narrowing. more chance of heart disease

The Goal of the analysis is to get a model that can predict with accuracy whether or not a person is likely to develop a heart disease considering the angiographic disease status.

Exploratory Data Analysis

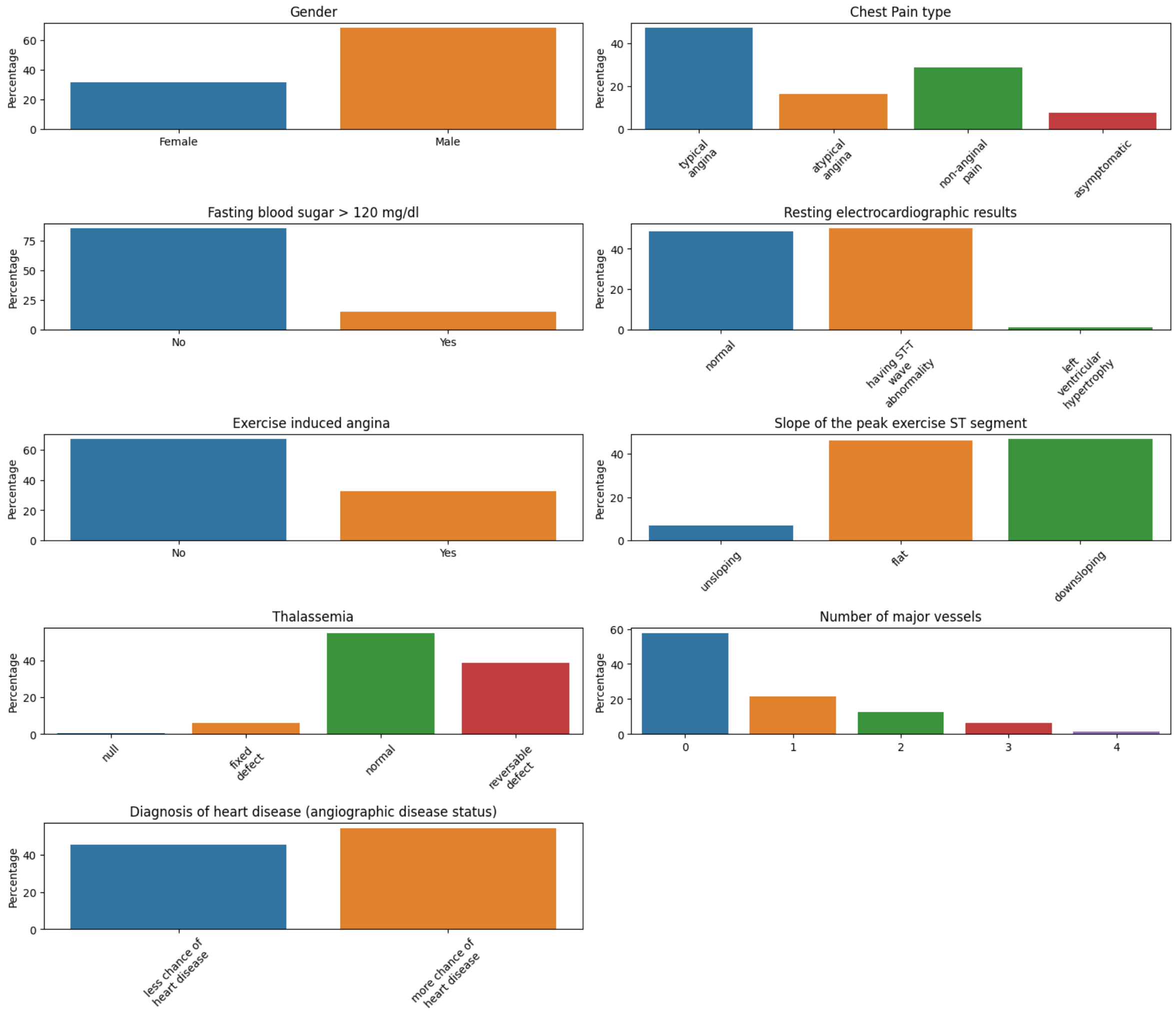
The first step was to load the csv file into a Pandas Dataframe. Then, the next step was to understand the data types of each features and deal with missing values. Luckily, the there wasn't any missing value. Also, it was important to identify numeric, ordinal, and categorical data. The features were classified as:

numeric = ['age', 'trtbps', 'chol', 'thalachh', 'oldpeak']

ordinal = ['caa']

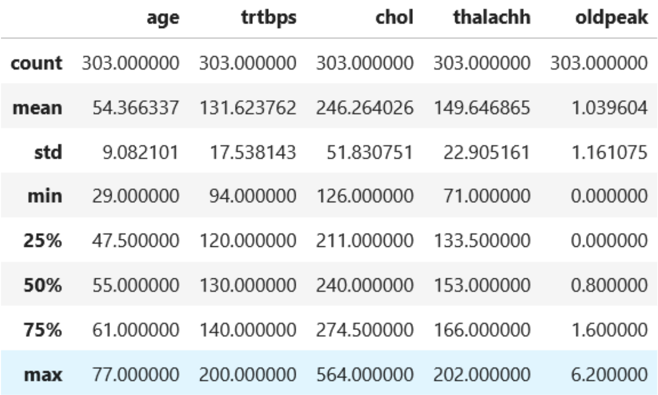
categorical = ['sex', 'cp', 'fbs', 'restecg', 'exng', 'slp', 'thall', 'output']

Next, univariate analysis was performed. For categorical variables, the bar plots are shown below.



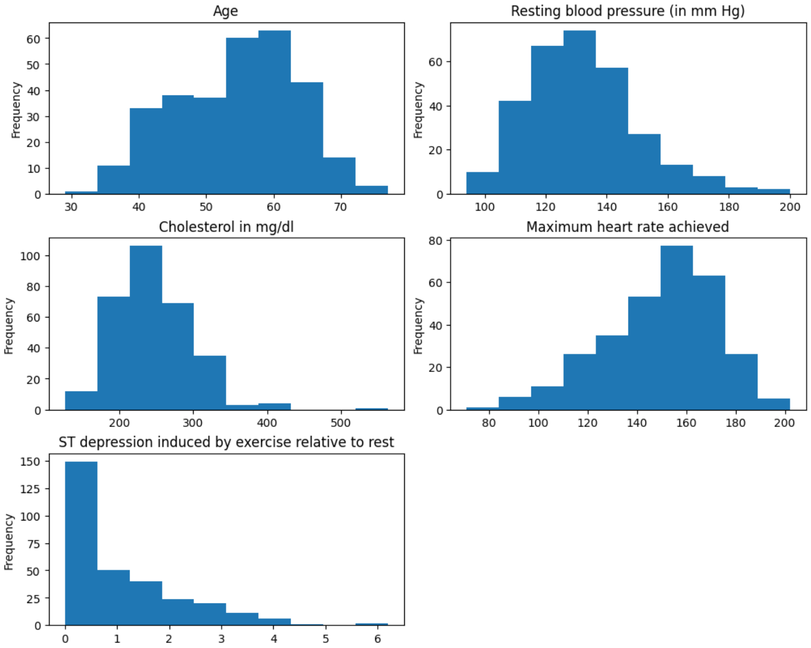
It can be seen that most of the patients are male. Also, typical angina, i.e. a type of chest pain that occurs when there is reduced blood flow to the heart muscle, is the most common type of chest pain. For most patients, blood sugar is below 120 mg/dl. Regarding to electrocardiographic results, the majority have ST-T wave abnormality. In addition, the majority of them doesn't have exercise induced angina and doesn't present thalassemia. Finally, the output, i.e. Diagnosis of heart disease (angiographic disease status), is relatively balanced with slightly more cases of "more chances of heart disease".

Then, the numerical variables were analyzed. First, a summary of important statistics was calculated:



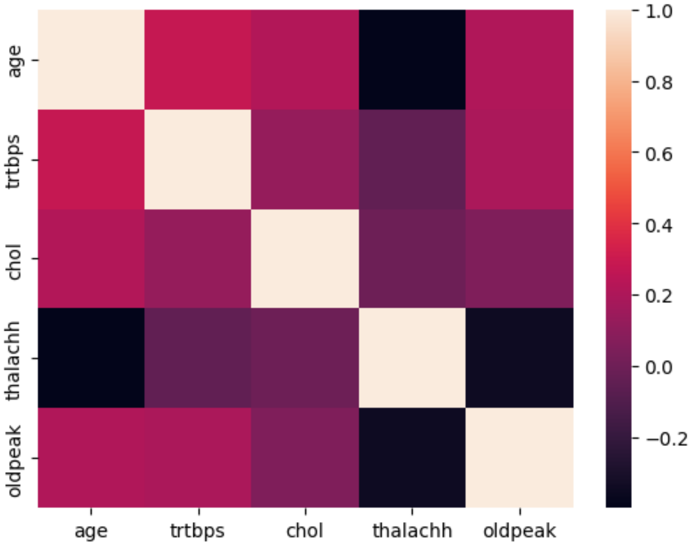
	age	trtbps	chol	thalachh	oldpeak
count	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	131.623762	246.264026	149.646865	1.039604
std	9.082101	17.538143	51.830751	22.905161	1.161075
min	29.000000	94.000000	126.000000	71.000000	0.000000
25%	47.500000	120.000000	211.000000	133.500000	0.000000
50%	55.000000	130.000000	240.000000	153.000000	0.800000
75%	61.000000	140.000000	274.500000	166.000000	1.600000
max	77.000000	200.000000	564.000000	202.000000	6.200000

It could be seen that there are 303 data points and that, on average, patients are 54 years old. The distribution of these variables are easier to understand with visualizations:



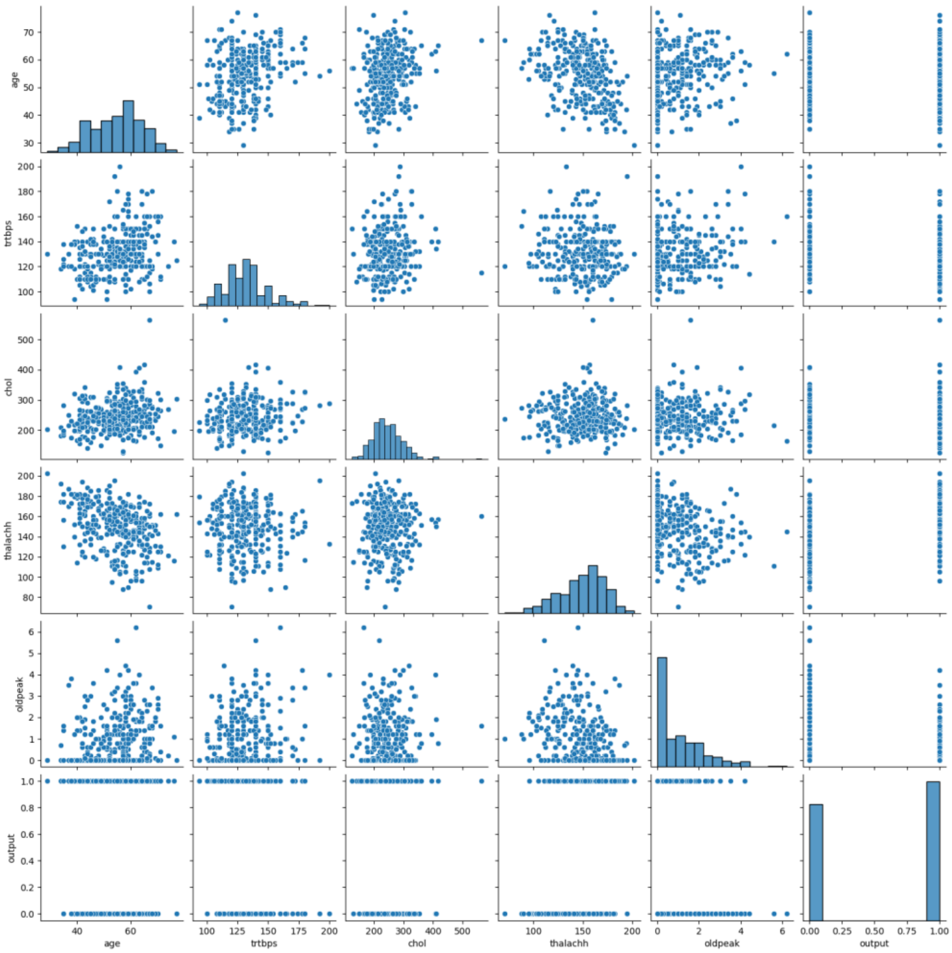
We can see that most of the patients are above 40 years old. Also, “Resting blood pressure” is right skewed with values up to 200 mmHg. “Maximum heart rate” is lightly left skewed and its values range from 80 up to 200. Finally, values for “ST depression induced by exercise” are mostly concentrated between 0 and 2, with some outliers.

The next step was to understand the correlations between the features. A heat map was displayed for this purpose:



We can see a high negative correlation between thalachh (maximum heart rate achieved) and age, and also between thalachh and oldpeak (ST depression induced by exercise relative to rest). In addition, positive correlation between trbps (resting blood pressure (in mm Hg)) and age is observed.

Finally, scatter plots between numerical variables, including the output, was obtained.



After a visual inspection, slight correlation could be seen between several variables. In the case of the output, none of the variables on their own seem to be strong predictors. For output prediction, it is necessary a multidimensional model.

Feature engineering steps were also performed. One hot encoding was used for categorical data, and Standard scaling was used for numerical data. The ordinal data was already encoded , with values ranging from 0 to 4. The final set of features was:

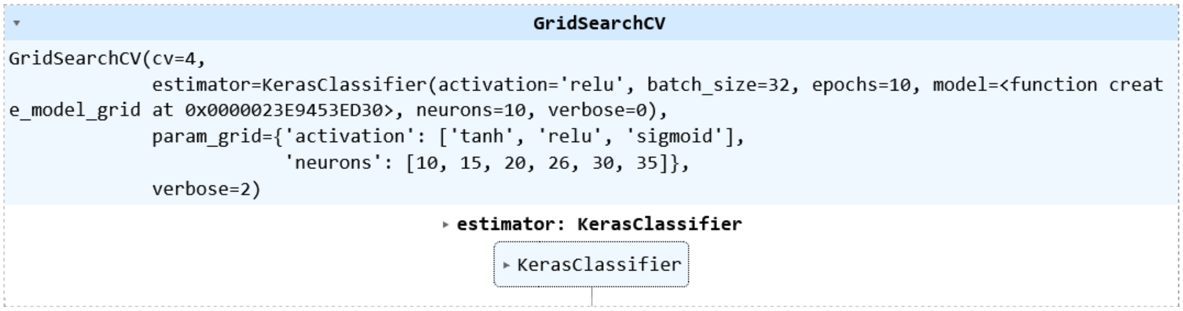
['age', 'trtbps', 'chol', 'thalachh', 'oldpeak', 'caa', 'sex_0', 'sex_1', 'cp_0', 'cp_1', 'cp_2', 'cp_3', 'fbs_0', 'fbs_1', 'restecg_0', 'restecg_1', 'restecg_2', 'exng_0', 'exng_1', 'slp_0', 'slp_1', 'slp_2', 'thall_0', 'thall_1', 'thall_2', 'thall_3']

Finally, the data was split in test-training sets, with 30% of the data in the test set.

Modeling

The model chosen for the classification task was the Multi-Layer Perceptron. Three different architectures were trained. The first with just one hidden layer, the second with two hidden layers, and the last one with three hidden layer. Grid search was used to get the best possible combination of hyper parameters. In order be able to use GridSearchCV function from Scikit-Learn, the library Scikeras (Adrian) was utilized.

The following are the parameters used for training the first model.



```
GridSearchCV(cv=4,
             estimator=KerasClassifier(activation='relu', batch_size=32, epochs=10, model=<function creat
e_model_grid at 0x0000023E9453ED30>, neurons=10, verbose=0),
             param_grid={'activation': ['tanh', 'relu', 'sigmoid'],
                         'neurons': [10, 15, 20, 26, 30, 35]},
             verbose=2)
```

The parameters used for the classifier were:

```
KerasClassifier(
        model=<function create_model_grid at 0x0000023E9453ED30>
        build_fn=None
        warm_start=False
        random_state=None
        optimizer=rmsprop
        loss=None
        metrics=None
        batch_size=32
        validation_batch_size=None
        verbose=0
        callbacks=None
        validation_split=0.0
        shuffle=True
        run_eagerly=False
        epochs=10
        activation=relu
        neurons=10
        class_weight=None
)
```

Where `<function create_model_grid at 0x0000023E9453ED30>` is the multi-layer perceptron model created in keras.

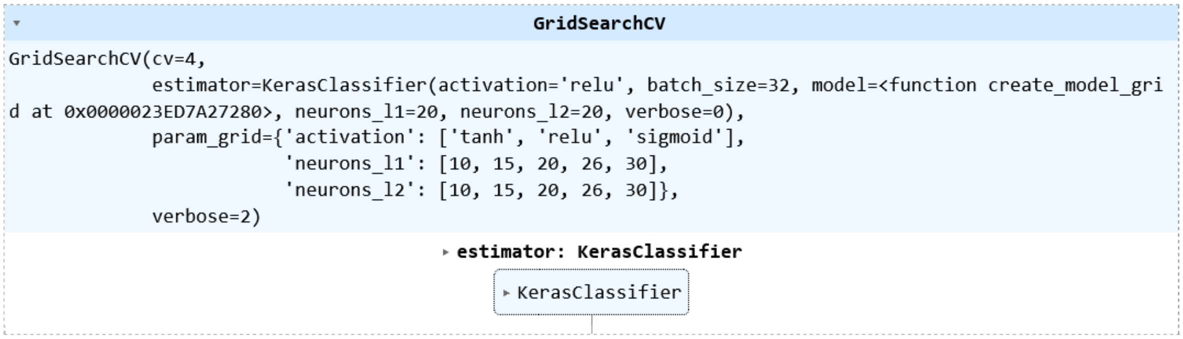
The best model found from the grid search was:

```
Best hyperparameters :  {'activation': 'sigmoid', 'neurons': 26}

Model: "sequential_3503"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 dense_10613 (Dense)         (None, 26)                702       

 dense_10614 (Dense)         (None, 1)                 27        

=================================================================
Total params: 729
Trainable params: 729
Non-trainable params: 0
_________________________________________________________________
```

With the following scores:

```
best cv score:  0.820754716981132
score for test set:  0.8131868131868132
```

The second architecture was the one with two hidden layers. The grid search object was:



```
GridSearchCV(cv=4,
             estimator=KerasClassifier(activation='relu', batch_size=32, model=<function create_model_gri
d at 0x0000023ED7A27280>, neurons_l1=20, neurons_l2=20, verbose=0),
             param_grid={'activation': ['tanh', 'relu', 'sigmoid'],
                         'neurons_l1': [10, 15, 20, 26, 30],
                         'neurons_l2': [10, 15, 20, 26, 30]},
             verbose=2)
```

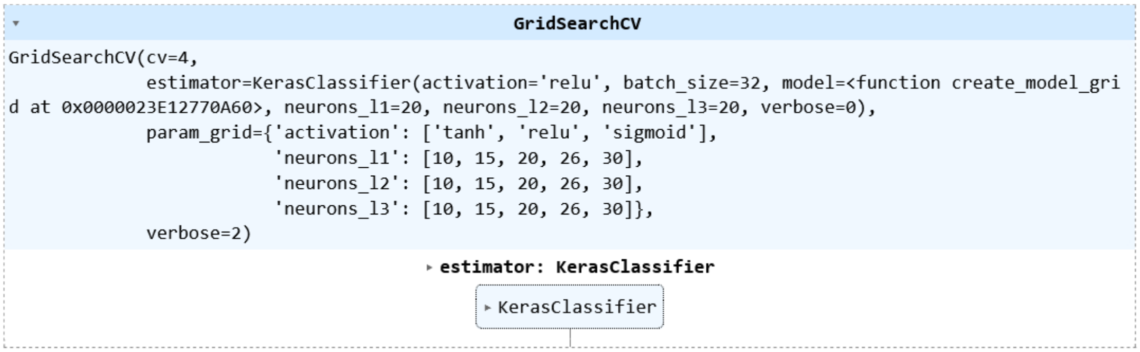
The best model found from this grid search was:

```
Best hyperparameters:  {'activation': 'tanh', 'neurons_l1': 26,
'neurons_l2': 15}

Model: "sequential_3430"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 dense_10466 (Dense)         (None, 26)                702       
                                                                 
 dense_10467 (Dense)         (None, 15)                405       
                                                                 
 dense_10468 (Dense)         (None, 1)                 16        
                                                                 
=================================================================
Total params: 1,123
Trainable params: 1,123
Non-trainable params: 0
_________________________________________________________________
```

The accuracy for this model was:

```
best cv score :  0.7405660377358492
score for test set :  0.7252747252747253
```

Finally, the third model had three hidden layers. The grid search object to find the optimal hyperparameters was:



```
GridSearchCV(cv=4,
             estimator=KerasClassifier(activation='relu', batch_size=32, model=<function create_model_gri
d at 0x0000023E12770A60>, neurons_l1=20, neurons_l2=20, neurons_l3=20, verbose=0),
             param_grid={'activation': ['tanh', 'relu', 'sigmoid'],
                         'neurons_l1': [10, 15, 20, 26, 30],
                         'neurons_l2': [10, 15, 20, 26, 30],
                         'neurons_l3': [10, 15, 20, 26, 30]},
             verbose=2)
```

The best model found from was:

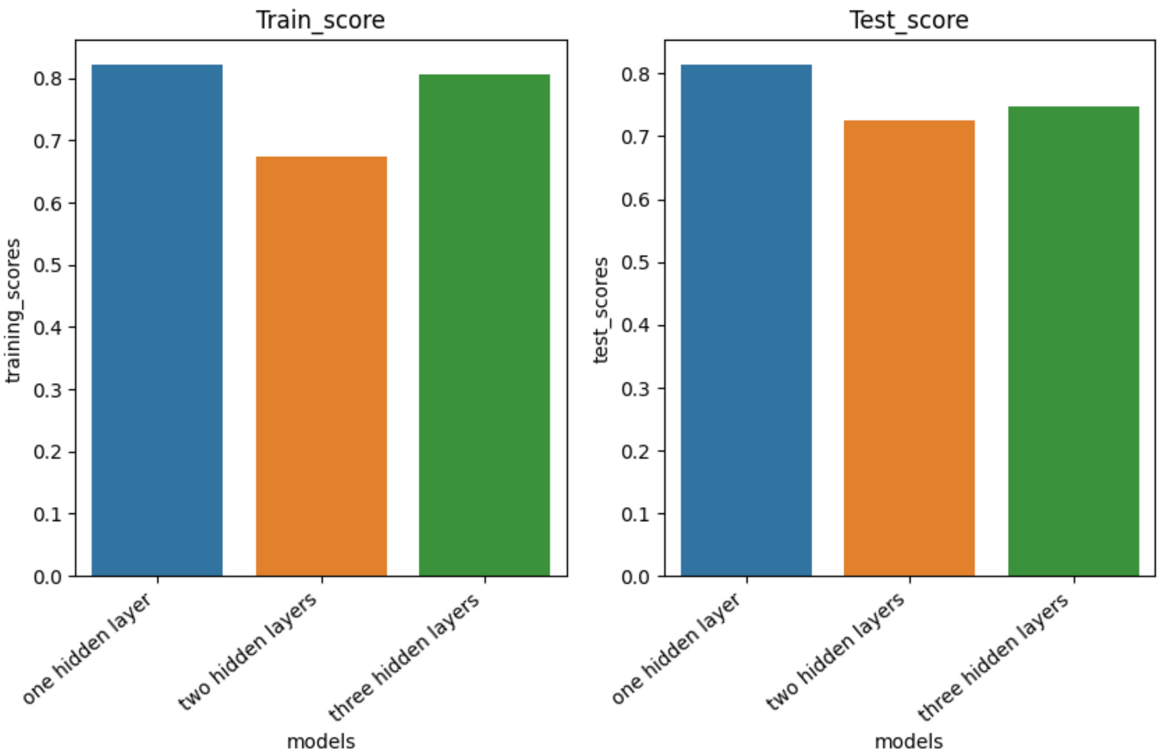
```
Best hyperparameters:  {'activation': 'tanh', 'neurons_l1': 26,
'neurons_l2': 20, 'neurons_l3': 30}


Model: "sequential_3007"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_9318 (Dense)          (None, 26)                702

 dense_9319 (Dense)          (None, 20)                540

 dense_9320 (Dense)          (None, 30)                630

 dense_9321 (Dense)          (None, 1)                 31

=================================================================
Total params: 1,903
Trainable params: 1,903
Non-trainable params: 0
_________________________________________________________________
```

The accuracy for this model was:

```
best cv score :  0.7971698113207548
score for test set :  0.7472527472527473
```

Final model

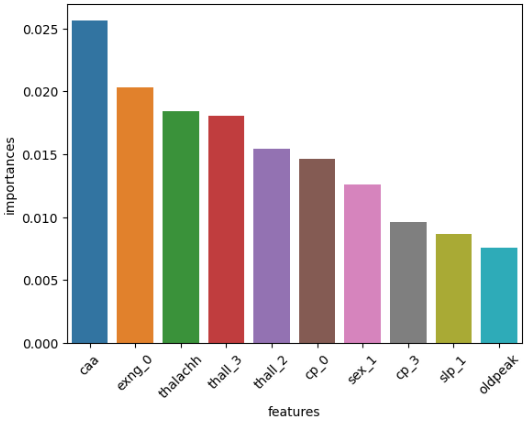
The focus is on accuracy, so the model to be selected needs to be the one with the highest prediction score. Below, barplots comparing the scores are shown:



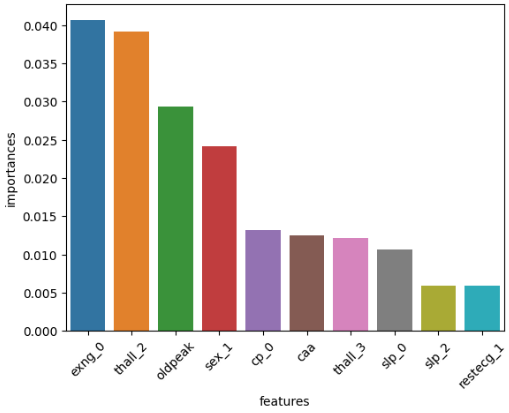
It could be seen that the architecture of one hidden layer is the one that performs the best, so this is the one chosen.

Finally, to satisfy interpretability requirements, an analysis of feature importance for this model was performed. The technique that was used for this purpose was permutation feature importance. Below are the results for the train and test set.

training set



Test set



It could be seen that for the test set, exng_0 (exercise induced angina = 0) is one of the most important features.

Key findings and insights

The best performing model was the simplest one. This can be explained using the complexity and the number of features of the data. For this model, the optimal number of hidden neurons was the same as the number of input neurons, and the activation function was sigmoid.

The most important feature when using the training set was “caa” (number of major vessels), but when using the test set, this feature is in the sixth position. This is suggesting that this feature may be causing some overfitting in the model. Other variable that could be causing overfitting is “thalachh” (maximum heart rate achieved) since it is in the third position when using the training set and it is not even in the first 10 positions when considering the training set.

For the test set, exng_0 (exercise induced angina = 0) is one of the most important feature for both, the training and test set. This is a strong predictor for the output and can be used for further domain specific studies. Being male, sex_01, also seem to be a strong predictor for developing heart disease.

The two categories thall_2 and thall_3 for the variable “thall” (thalassemia) are important in both the training and test set, which is also a variable to take into account in future studies.

Suggestions and next steps

It is possible to perform an exhaustive grid search with more architectures and a variety of hyper parameters to find better performing models. Also, the use of different evaluation metric can be very valuable when assessing these model. Metrics such as f-score, precision, recall can be used for this purpose.

Also, a study on the variables that are sources of overfitting, such as “caa” or “thalachh”, could be perform in order to assess the causes of this. In addition, further research regarding the most important features is suggested. This could be a causality study to see if there is a direct cause-effect or if there exists a third confounding variable. If the cause-effect relationship exists, medical research could be performed to understand this relationship.

With the additional data that is going to be capture at the moment of patient registration, a new model could be developed. This new model can capture the characteristics of this specific population and have an improve prediction power.

References

Adrian Garcia Badaracco. 12 dic 2022. SciKeras. MIT License (MIT).
www.adriangb.com/scikeras/stable/

Rashik Rahman. 2021. Heart Attack Analysis & Prediction Dataset. Kaggle.
www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset