Faculty of Engineering & Technology
Electrical & Computer Engineering Department

MACHINE LEARNING AND DATA SCIENCE – ENCS5341

Assignment#3 Report

Prepared by:
Oday Ziq

Section: 1

Date: 26/1/2023

Introduction

The main goal of this project is to develop a predictive model for heart disease, by using various machine learning algorithms. Heart disease is one of the most common chronic diseases around the world, and is considered one of the main causes of death. The early and accurate prediction of such that disease can be very important in receiving the necessary treatment early. By using a dataset that contains patient medical information, this project aims to apply different machine learning models to predict the presence of heart disease, in addition to analyze those models and compare their performance.

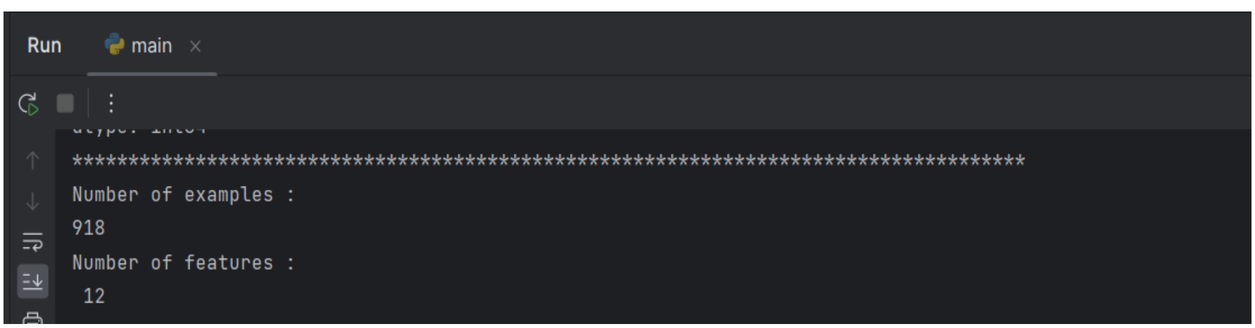
Dataset

The used dataset is a collection of patient records which aims to predict the presence of heart disease. The heart disease dataset contains multiple features, which are: Age which represents the patient's age in years, Sex which represents the patient's gender (Male or Female), Chest pain type which represents the categorization of chest pain experienced by the patient (ATA, NAP, ASY, TA), Resting Blood Pressure which represents the blood pressure of the patient when rest, Fasting Blood Sugar which represents the level of the blood sugar when fasting, Resting Electrocardiogram Results, Maximum Heart Rate, Exercise-Induced Angina, ST Depression which is the depression that is induced by exercise relative to rest and the ST Slope which is the slope of the peak exercise ST segment.

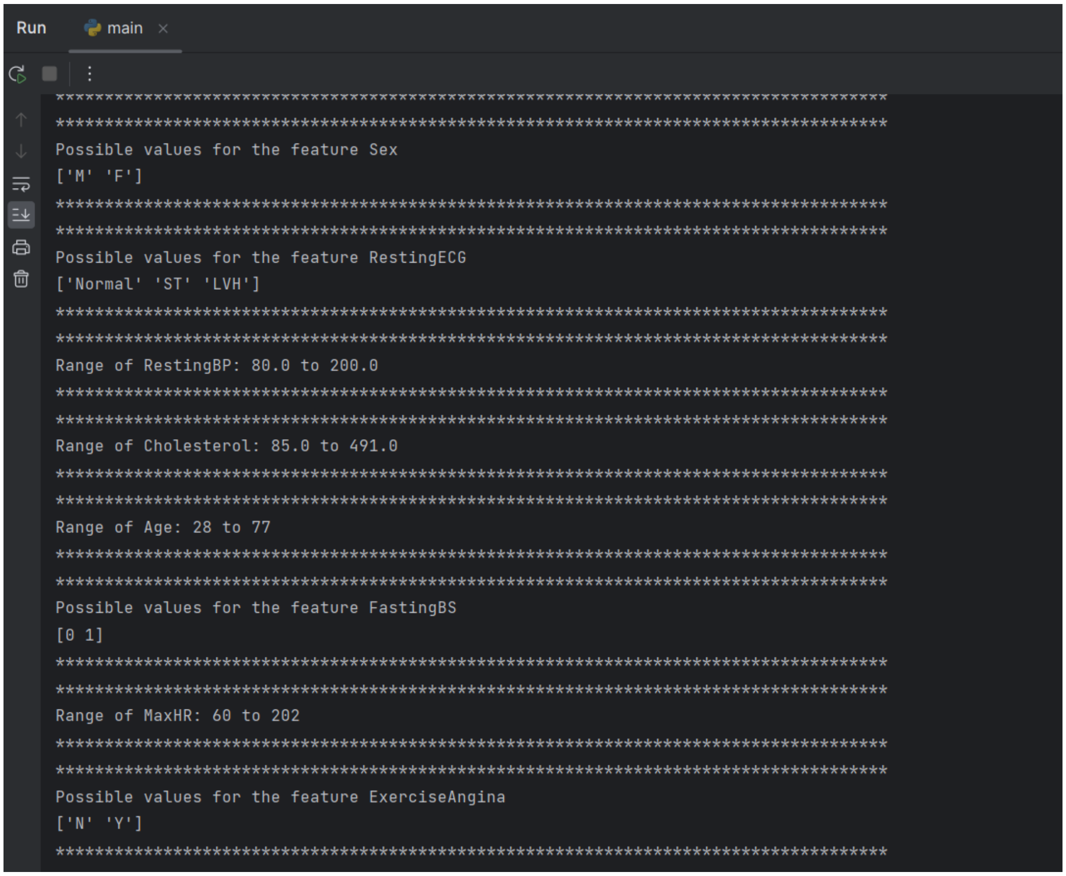
FastingBS value is 1 when the blood sugar is greater than 120 mg/dl, and 0 otherwise. The values for Resting Blood Pressure are: TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic. The values for ExerciseAngina are: Y(yes) if the patient has the Exercise-induced angina, and N(no) if the patient doesn't suffer from it. The ST_Slope values are: Up: upsloping, Flat: flat, Down: downsloping.

The target variable of this dataset is the presence or absence of heart disease (HeartDisease), which is encoded as 1 for presence and 0 for absence.

The following figure shows the output for the number of examples and features in the dataset:

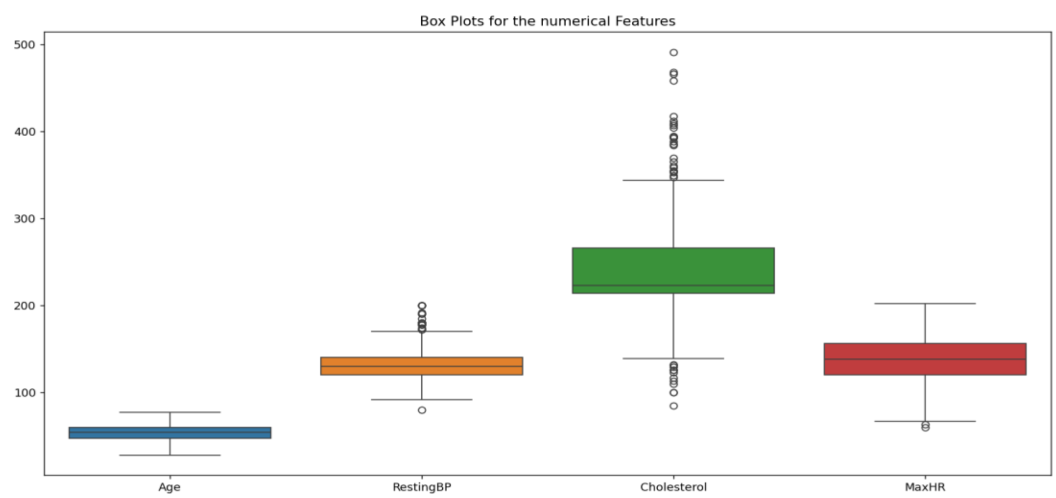


The following figure shows the possible values for the categorical data and the ranges for the numeric of some features:

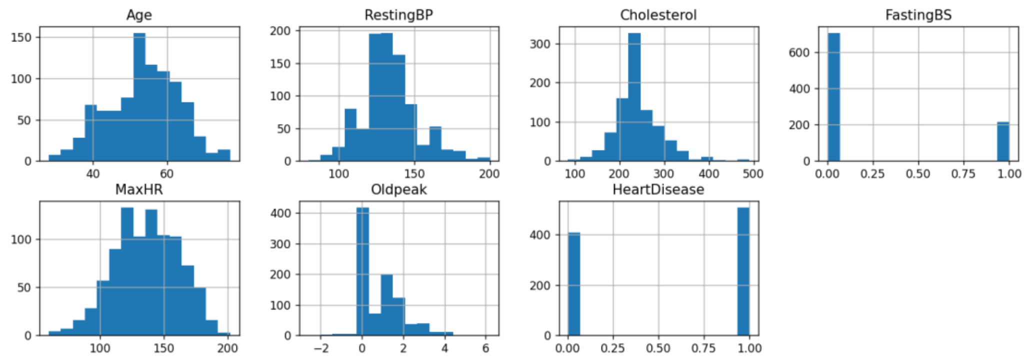


```
Run   main ×
*************************************************************************************
*************************************************************************************
Possible values for the feature Sex
['M' 'F']
*************************************************************************************
*************************************************************************************
Possible values for the feature RestingECG
['Normal' 'ST' 'LVH']
*************************************************************************************
*************************************************************************************
Range of RestingBP: 80.0 to 200.0
*************************************************************************************
*************************************************************************************
Range of Cholesterol: 85.0 to 491.0
*************************************************************************************
*************************************************************************************
Range of Age: 28 to 77
*************************************************************************************
*************************************************************************************
Possible values for the feature FastingBS
[0 1]
*************************************************************************************
*************************************************************************************
Range of MaxHR: 60 to 202
*************************************************************************************
*************************************************************************************
Possible values for the feature ExerciseAngina
['N' 'Y']
*************************************************************************************
```

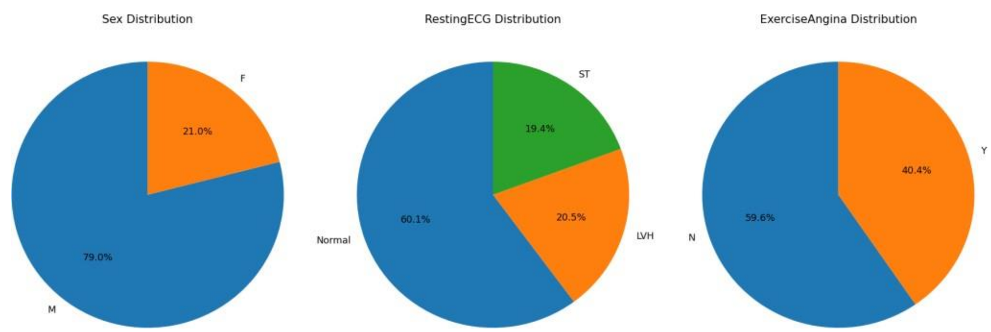
The following figure shows the box plots for some numerical features:



A histogram was plotted to all the numerical features, in order to see the distribution of each feature among the possible values, as shown in the figure below:



For the categorical features, a pie chart was plotted for each feature, in order to see the proportion formed by each value of these features:



It has been noticed that there is 21% of the patients are females, and 79% are males. Also there 19.4% of the patients have a RestingECG of ST type, 20.5% of LVH type, and 60.1% of normal type. In addition to that, there are 59.6% of the patients don't suffer from the ExerciseAngina, while 40.4% are suffering from it.

It has been noticed that there are no missing values in the features, as the output below shows:

```
Run   main
C:\Users\user\PycharmProjects\MLproject\venv\Scripts\python.exe C:\Users\user\PycharmProjects\MLproject\main.py
*************************************************************************************
The number of missing values for each feature is :
Age               0
Sex               0
ChestPainType     0
RestingBP         0
Cholesterol       0
FastingBS         0
RestingECG        0
MaxHR             0
ExerciseAngina    0
Oldpeak           0
ST_Slope          0
HeartDisease      0
dtype: int64
```

A descriptive statistics like count, mean and other statistics were applied to the dataset also, as the output below shows:

```
Run   main ×

************************************************************************************

Here are descriptive statistics of the dataset

              Age   RestingBP  Cholesterol   FastingBS       MaxHR     Oldpeak
count  918.000000  918.000000   918.000000  918.000000  918.000000  918.000000
mean    53.510893  132.538126   239.141612    0.233115  136.809368    0.887364
std      9.432617   17.990127    49.814548    0.423046   25.460334    1.066570
min     28.000000   80.000000    85.000000    0.000000   60.000000   -2.600000
25%     47.000000  120.000000   214.000000    0.000000  120.000000    0.000000
50%     54.000000  130.000000   223.000000    0.000000  138.000000    0.600000
75%     60.000000  140.000000   266.000000    0.000000  156.000000    1.500000
max     77.000000  200.000000   491.000000    1.000000  202.000000    6.200000
```

Experiments and Results

In order to deal with this task, we have used several machine learning models that we have learnt.

k-Nearest Neighbors (kNN)

As a baseline model, the kNN classifier was experimented, with adifferent values of k (k=1 and k=3), where k represents the number of nearest neighbors considered for classification. The dataset was split into training set and testing set in order to evaluate the kNN models. Different metrics were used in order to evaluate this model, like accuracy, precision, recall, F1-score, and the confusion matrix. The output observed by the kNN model was as shown below:

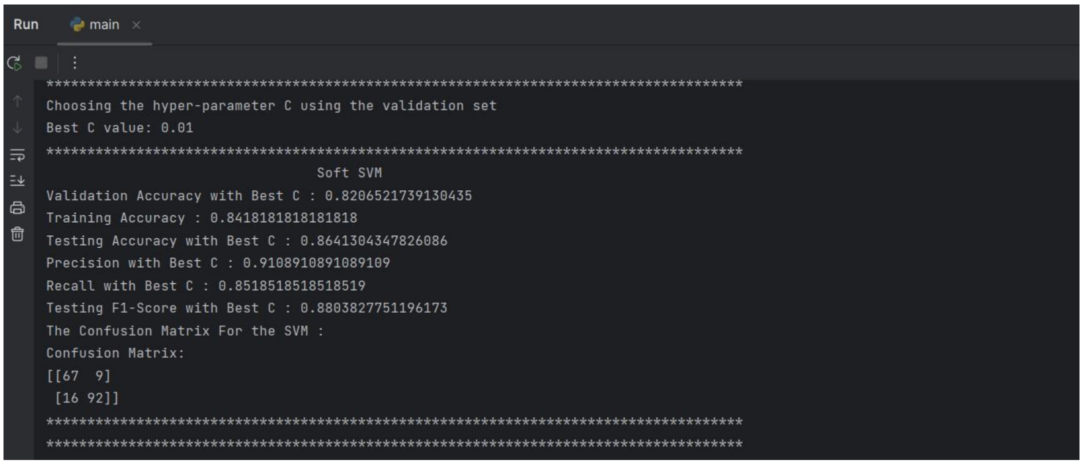
```
Run   main ×

************************************************************************************
                                 kNN
Evaluation Metrics For k=1
Confusion Matrix:
[[ 94  18]
 [ 30 134]]
Training Accuracy (k=1):  1.0
Testing Accuracy (k=1):  0.8260869565217391
Precision (k=1):  0.881578947368421
Recall (k=1):  0.8170731707317073
F1 Score (k=1):  0.8481012658227848
************************************************************************************
************************************************************************************
                                 kNN
Evaluation Metrics For k=3
Confusion Matrix:
[[ 95  17]
 [ 30 134]]
Training Accuracy (k=3):  0.8847352024922118
Testing Accuracy (k=3):  0.8297101449275363
Precision (k=3):  0.8874172185430463
Recall (k=3):  0.8170731707317073
F1 Score (k=3):  0.8507936507936508
************************************************************************************
```

For the kNN with k=1, the testing accuracy was 82.6% and F1-score of 84.81%, whereas when k=3, the testing accuracy was 82.97% and F1-score of 85.07%. Both kNN models show a good performance, with k=3 more suitable due to its high accuracy and lower probability of overfitting.

Soft Margin Support Vector Machines (SVM)

In order to achieve a better performance, the soft margin support vector machine was experimented. To find the hyper-parameter (C), some experiments were conducted using the cross validation among the validation set in order to select the best value that gives the highest validation accuracy, the dataset was split into training set, validation set and testing set in order to evaluate the SVM model. The Soft SVM showed good result, and it was evaluated using the metrics accuracy, precision, recall, F1-score, and the confusion matrix, as the output below shows:



```
Run   main ×
*************************************************************************************
Choosing the hyper-parameter C using the validation set
Best C value: 0.01
*************************************************************************************
                                 Soft SVM
Validation Accuracy with Best C : 0.8206521739130435
Training Accuracy : 0.8418181818181818
Testing Accuracy with Best C : 0.8641304347826086
Precision with Best C : 0.9108910891089109
Recall with Best C : 0.8518518518518519
Testing F1-Score with Best C : 0.8803827751196173
The Confusion Matrix For the SVM :
Confusion Matrix:
[[67  9]
 [16 92]]
*************************************************************************************
*************************************************************************************
```

The Soft SVM model with the selected hyper-parameter C value of 0.01 shows a good performance on the test set, such that it gives a testing accuracy of 86.41%, precision of 91.08%, and F1-score of 88.03%. Also, the confusion matrix obtained by the SVM was as follows:

[[TN:67 FP:9]

[FN:16 TP:92]]

The confusion matrix obtained by the SVM shows 92 True Positive classified examples from the testing set, 67 True Negative classified examples, 9 False Positive classified examples and 16 False Negative classified examples.

Random Forest

In the Random Forest classifier, an ensemble learning was applied, which uses multiple decision trees to make predictions. Different values of the hyper-parameter (n_estimators) which were experimented among a validation set in order to select the best value that gives the highest validation accuracy, the dataset was split into training set, validation set and testing set in order to evaluate the Random Forest model. The Random Forest showed good result, and it was evaluated using the metrics accuracy, precision, recall, F1-score, and the confusion matrix, as the output below shows:

```
Run    main ×

***********************************************************************************
***********************************************************************************
                                 Random Forest
Best Hyperparameters: {'n_estimators': 200}
Training Accuracy with Best Model: 1.0
Testing Accuracy with Best Model: 0.8858695652173914
Precision with Best Model: 0.9393939393939394
Recall with Best Model: 0.8611111111111112
Testing F1-Score with Best Model: 0.8985507246376812
The Confusion Matrix For the Random Forest :
Confusion Matrix:
[[70  6]
 [15 93]]
***********************************************************************************
```

Overall, the Random Forest model performs well, and it achieves a high testing accuracy of 88.58%, and a precision of 93.93%, and F1-score of 89.85%. The confusion matrix obtained by the Random Forest was as follows:

[[TN:70 FP:6]

[FN:15 TP:93]]

The confusion matrix obtained by the Random Forest shows 93 True Positive classified examples from the testing set, 70 True Negative classified examples, 6 False Positive classified examples and 15 False Negative classified examples.

Analysis

The analysis of the models begin with the selection of an appropriate hyper-parameters using the validation set. Each model was trained on a training dataset and evaluated on a separate testing dataset, which ensures that the model's performance is assessed on an unseen data, and this provides the capability of the model to generalize. For the evaluation, various evaluation metrics were used in order to assess the model's performance, the used metrics included precision, recall, accuracy, F1-score, and the confusion matrix.

There are many interesting findings that were noticed in the analysis of the models, like choosing the hyper-parameters and how it can influence the model's performance. Both the SVM and Random Forest models demonstrate strong classification performance with a relatively low number of false positives and false negatives, and when these two models were compared, it has been noticed that the best model was the Random Forest, such that it has the highest testing accuracy and the lowest misclassified number of examples, as the output below shows:

```
***************************************************************
                                  Soft SVM
Validation Accuracy with Best C : 0.8206521739130435
Training Accuracy : 0.8418181818181818
Testing Accuracy with Best C : 0.8641304347826086
```

For the figure above, the highlighted accuracy represents the testing accuracy for the Soft SVM model which is approximately 86.41%.

[[TN:67 FP:9]

[FN:16 TP:92]]

The confusion matrix above, shows the classified examples for the Soft SVM model, with a total number of 25 misclassified examples.

```
*************************************************************************
                                 Random Forest
Best Hyperparameters: {'n_estimators': 200}
Training Accuracy with Best Model: 1.0
Testing Accuracy with Best Model: 0.8858695652173914
```

For the figure above, the highlighted accuracy represents the testing accuracy for the Random Forest model which is approximately 88.58%.

[[TN:70 FP:6]

[FN:15 TP:93]]

The confusion matrix above, shows the classified examples for the Random Forest model, with a total number of 21 misclassified examples.

It has been noticed that the best model was the Random Forest and that's because it has the highest testing accuracy (88.58%) and the lowest number of misclassified examples (21 example).

For the Random Forest classifier, the indices of the misclassified examples were observed by comparing the actual value of the target variable and the predicted value, and detect any mismatch. As mentioned before, the misclassified examples for the Random Forest were 21 examples, as the output below shows:

```
Run    main ×

*************************************************************************************
Indices of the Misclassified examples by the random forest classifier :
[  5  15  26  28  32  46  53  55  61  65  68  83 107 122 129 133 142 161
 169 174 180]
```

As shown above, the misclassified examples were 21.

Conclusions and Discussion

It has been noticed that better performance was achieved after evaluating the SVM and the Random Forest, such that the highest testing accuracy when using the kNN classifier was when k=3, which is 82.97%, whereas the testing accuracy in SVM was 86.41% and Random Forest was 88.58%. The improvement of performance may refer to several reasons, like the model complexity, such that kNN is simple learning algorithm that relies on the entire dataset for predictions, and it doesn't explore the complex relationship between features, whereas SVM and Random Forest are more complex models, and they have the ability of exploring the relationships between the Features. In addition to that, kNN can suffer from the problem of Curse of Dimensionality, and it can be sensitive to the noisy data.

The analysis has shown that both the Soft SVM and the Random Forest models have the ability of achieving strong classification performance, such that they demonstrated high accuracy in the classification.

Both models have limitations, such that the SVM's performance relies on the parameter tuning, which can be expensive from the side of computations for large datasets. Random Forest also can have limitations, such that more tress slow down the model.

The choice of evaluation metrics should follow the specific goals of the classification task, such that the accuracy provides an overall measure of correctness, precision and recall are critical in scenarios where minimizing false positives or false negatives is essential. The F1-score balances both precision and recall, offering a more comprehensive assessment.