# AML Final Project

# Predicting Heart Attack Disease

Submission for Applied Machine Learning CH

Date of submitting: 9/2/2022


Prepared by:

Afraa Farhan: 1808552

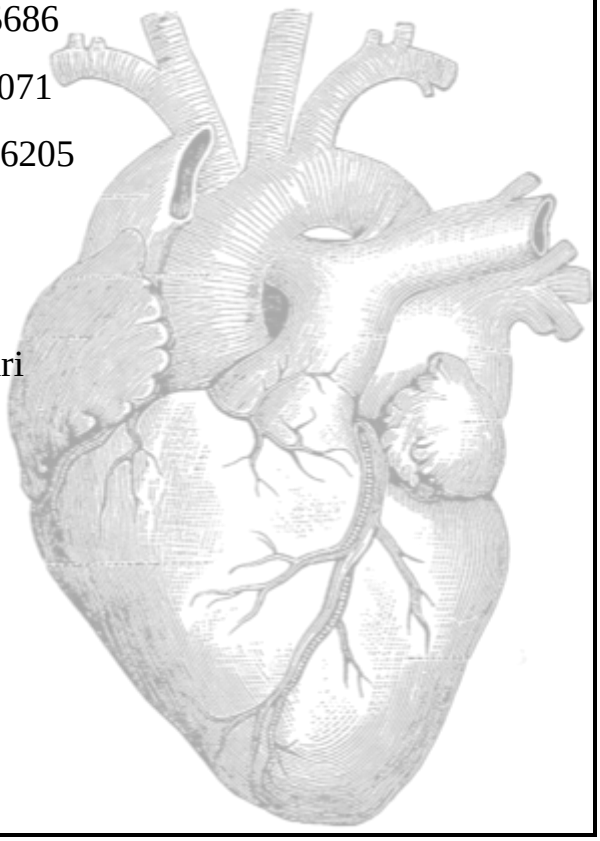
Manar Bagabas: 2005686

Joud Alahmari: 2008071

Mashael Alshehri: 2006205

Prepared for:

Dr. Lobna Alhassairi

## 1. Choose an investigation (scenario) and identify pre-existing sources of data that can address a particular machine learning goal.

**Choose, and state, the goal, and reasons why the datasets were chosen and how they were found.**

a. The goal of this data analysis is to understand the relationships and patterns between different features of heart disease patients and to predict the likelihood of heart disease in new patients.

b. The datasets ('heart.csv') were chosen because heart disease is a major health issue and is one of the leading causes of death globally. This dataset provides valuable information on various health and lifestyle factors of patients, including their age, sex, cholesterol levels, blood pressure, and other important parameters. The data in this dataset can be used to build predictive models that can help healthcare professionals diagnose and treat heart disease more effectively.

c. This dataset was found through various online open-source repositories such as UCI Machine Learning Repository and Kaggle, which host datasets from various domains, including healthcare. The data in these repositories is curated and validated, making them suitable for data analysis and machine learning projects.

**Develop and state one or more question/ hypothesis related to the goal of the investigation and that can be answered using the datasets under consideration.**

The goal of this investigation is to identify the factors that contribute to heart disease and to develop a model that can predict the presence of heart disease.

**One question** that can be answered using this dataset is:

- What are the most significant factors that contribute to the development of heart disease?

**To answer this question**, we can use the correlation matrix method to identify which features have the strongest relationship with the target variable, which represents the presence or absence of heart disease. By examining the correlations between the features and the target, we can gain insights into which factors are most strongly associated with heart disease and which are less important.

**Another hypothesis** that can be tested using this dataset is:

- Can a machine learning model be developed that accurately predicts the presence of heart disease based on a patient's medical history and other factors?

**To test this hypothesis**, we can use the support vector machine (SVM) algorithm to train a model on the heart dataset and evaluate its performance using a performance metric like accuracy. If the accuracy of the model is high, it would suggest that a machine learning model can be effective in predicting the presence of heart disease based on the patient's medical history and other factors.

**Provide a description of the choice of tools\libraries used.**

The choice of tools/libraries used depends on the specific requirements of the problem and the data analysis task being performed. In this case, we will be using Python programming language and its libraries such as Pandas, Numpy, Matplotlib and Scikit-Learn.

- **Pandas** is a library used for data manipulation and analysis. It provides data structures for efficiently storing large datasets and tools for working with them.
- **Numpy** is a library for numerical computing in Python. It provides support for arrays and matrices, which are important for efficient computation in many machine learning algorithms.
- **Matplotlib** is a plotting library for Python. It provides support for creating various types of plots and charts, which can be used for visualizing the data and understanding the relationships between features.
- **Scikit-Learn** is a machine learning library for Python. It provides a range of algorithms for supervised and unsupervised learning, including support vector machines (SVM), which we will be using in this case to build our model.

These libraries have been chosen due to their ease of use, robustness, and the wide range of functionality they offer for data analysis and machine learning tasks.

## 2. Data Analysis

**Design an analysis study to answer the above questions and document the analysis design.**

In this analysis study, the goal is to predict whether a person has heart disease or not based on various features such as age, sex, cholesterol levels, etc. The dataset 'heart.csv' was selected for this analysis as it contains patient data with a target column indicating whether the person has heart disease or not, which makes it ideal for a binary classification problem.

- The **first step** in the analysis was to import the necessary libraries, such as Numpy, Pandas, Matplotlib, and Seaborn. The dataset was then loaded into a Pandas data frame.
  This is how our dataset looks like:

| | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca | thal | target |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 52 | 1 | 0 | 125 | 212 | 0 | 1 | 168 | 0 | 1.0 | 2 | 2 | 3 | 0 |
| 1 | 53 | 1 | 0 | 140 | 203 | 1 | 0 | 155 | 1 | 3.1 | 0 | 0 | 3 | 0 |
| 2 | 70 | 1 | 0 | 145 | 174 | 0 | 1 | 125 | 1 | 2.6 | 0 | 0 | 3 | 0 |
| 3 | 61 | 1 | 0 | 148 | 203 | 0 | 1 | 161 | 0 | 0.0 | 2 | 1 | 3 | 0 |
| 4 | 62 | 0 | 0 | 138 | 294 | 1 | 1 | 106 | 0 | 1.9 | 1 | 3 | 2 | 0 |

- After that we check the missing values were checked. Since there were no missing values, this dataset was considered suitable for further analysis.

```
#Checking the missing values
data.isnull().any()

# The answer shows that there is no missing values so one of the reason to select this data
```

```
age         False
sex         False
cp          False
trestbps    False
chol        False
fbs         False
restecg     False
thalach     False
exang       False
oldpeak     False
slope       False
ca          False
thal        False
target      False
dtype: bool
```

- Also we check and analyses the information about the data types and count of the columns of the dataset

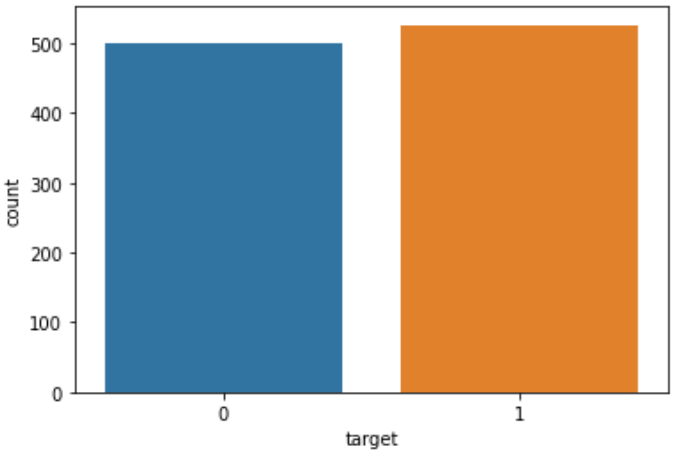
```
# Get information about the dataframe, including column names and data types
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1025 non-null   int64  
 1   sex       1025 non-null   int64  
 2   cp        1025 non-null   int64  
 3   trestbps  1025 non-null   int64  
 4   chol      1025 non-null   int64  
 5   fbs       1025 non-null   int64  
 6   restecg   1025 non-null   int64  
 7   thalach   1025 non-null   int64  
 8   exang     1025 non-null   int64  
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64  
 11  ca        1025 non-null   int64  
 12  thal      1025 non-null   int64  
 13  target    1025 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
None
```

After that, **Exploratory Data Analysis** was performed to understand the data better.

- It involved checking the count of people who have and do not have heart disease. The count plot and print statements showed that the data is balanced, which is an ideal scenario for a machine learning model.

```
# Explratory Data Analysis

# 1. Checking the count of how many people are having or not having heart disease
sns.countplot(x='target', data = data)
plt.show()
```

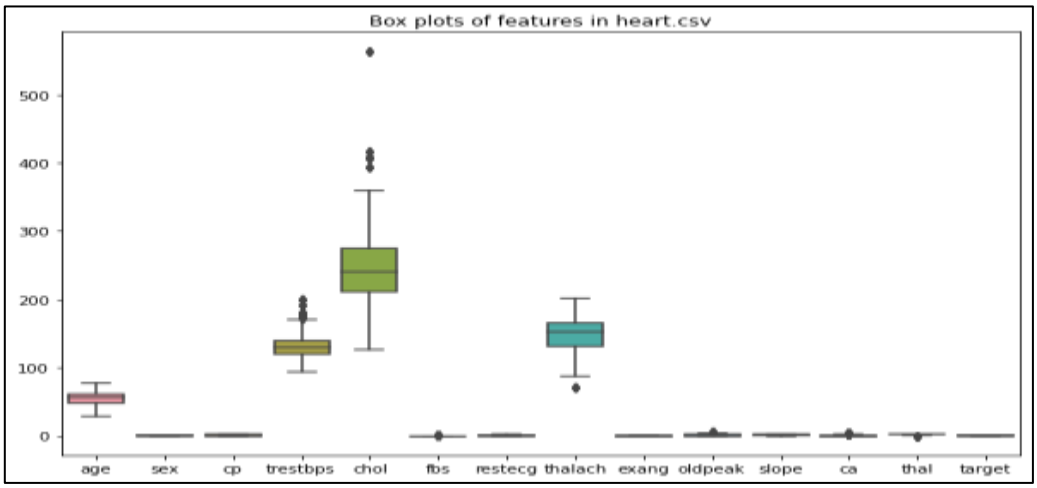


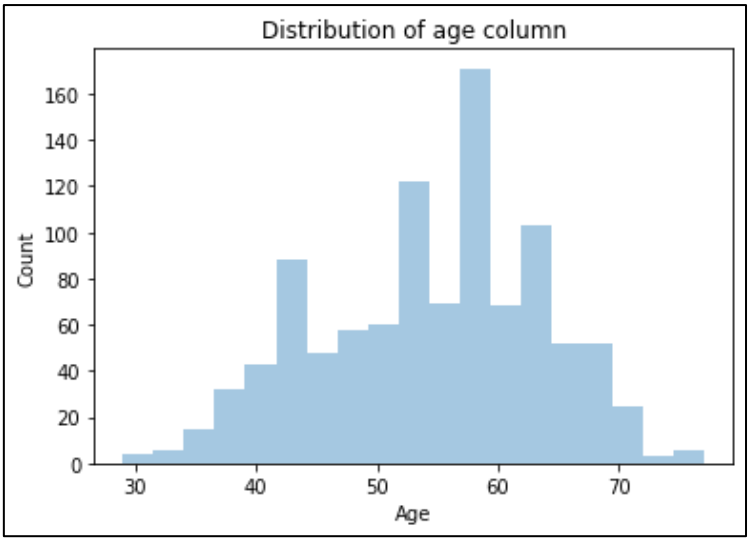

```
print('Having disease = ',(data['target'] == 0).sum())
print('Not Having disease = ',(data['target'] == 1).sum())

# The plot and count also shows us one more thing that the data is prety much balanced
```

```
Having disease =  499
Not Having disease =  526
```

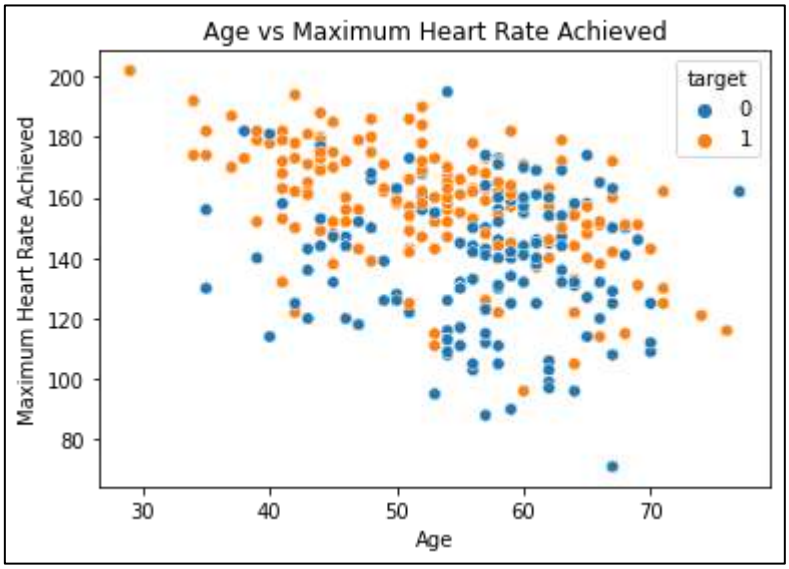
In the next step, **Feature Engineering** was performed to understand features more closely.

- We plot a boxplot which is used to visualize the distribution of the features. Also, the distribution of age column is plotted by using histogram.

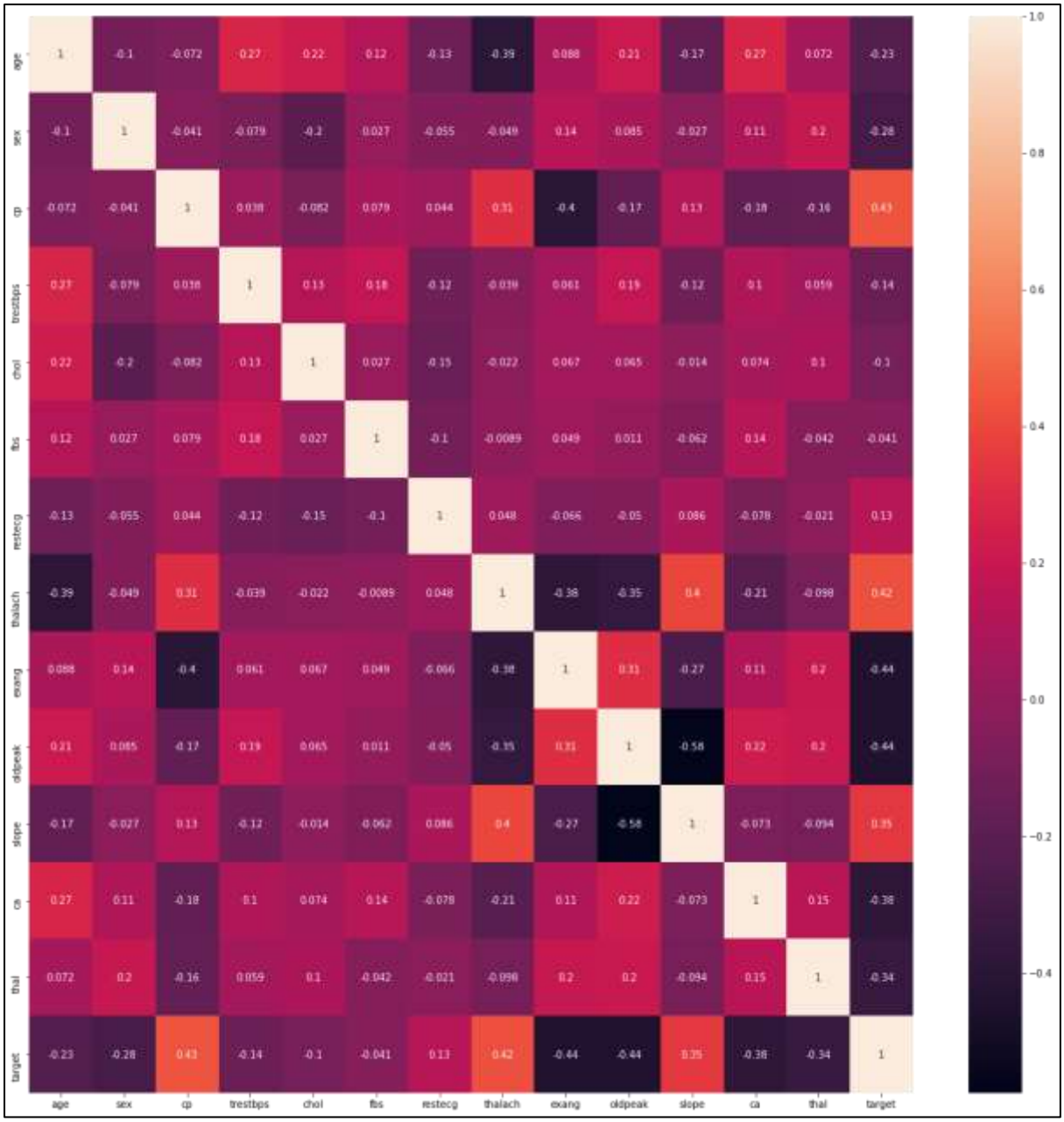





- Scatterplot of age versus maximum heart achieved is plotted to see the in-depth relationship between the two features.



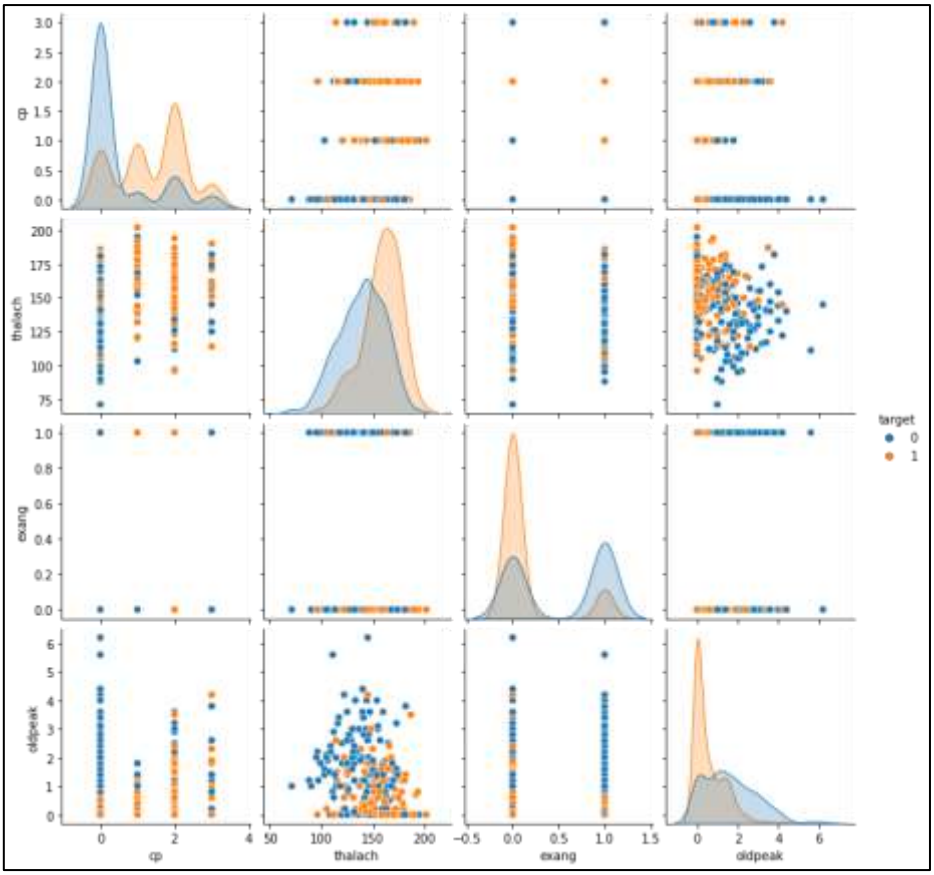

- A correlation matrix was created to check the relationship between the features and the target and plotted as a heatmap. This is important because it provides us with a visual representation of the relationship between different features and enables us to understand which features are positively or negatively correlated.




- Based on a threshold of 0.4 and -0.4, high correlated features are selected. A pair plot is then plotted to visualize the relationships between these features. This is important because it enables us to see the relationships between the selected features and understand if there is any linear or non-linear relationship between them.

```
# Checking the feautures having high correlation
high_corr=[]
mat= correlation_matrix.iloc[:-1]["target"]
for index,i in enumerate(mat):
    if i >= 0.4 or i <= -0.4:
        high_corr.append(mat.index[index])
print('Feaures having High Correlation are: \n',high_corr)
```

```
Feaures having High Correlation are:
 ['cp', 'thalach', 'exang', 'oldpeak']
```




**Perform the machine learning algorithms and identify the most suitable algorithm(s) in a form that can be validated and describe the steps and results you took to ensure this validation.**

---

For this dataset the machine learning model is trained using a **Support Vector Machine (SVM)** classifier.

There are other machine learning techniques that could be applied to this dataset, such as decision trees, random forests, k-nearest neighbors (KNN), and logistic regression, among others.

SVM is preferred over these techniques for the following reasons:

- SVM can handle complex datasets and is effective in dealing with high-dimensional data, making it well-suited for this particular dataset.
- SVM is good for datasets where there is a clear separation between the classes, which is the case in this dataset.
- SVM can handle non-linearly separable data through the use of kernels, which is useful in situations where the classes in the dataset are not linearly separable.
- SVM is less susceptible to overfitting compared to other techniques, such as decision trees, which can result in a high variance model that is not generalizable to new data.
- SVM has been found to produce robust and accurate results for many classification problems, making it a popular choice among machine learning practitioners.

The following are the **steps of how we implement the SVM machine learning model**:

**Splitting the Dataset:** The data is split into target and features. The target column is assigned to the Y variable and the rest of the features are assigned to the X variable. The data is then split into training and test data using *train_test_split* from scikit-learn with a test size of 0.33.

```
# Now we are going to implement the Machine Learning Model

# Splittng the Dataset into target and features
X = data.drop("target", axis = 1)
Y = data["target"]
```

```
# Splittng the Dataset into training and test data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.33, random_state=0)
```

**Training the Classifier:** The code uses Support Vector Machine (SVM) classifier from scikit-learn to train the model. The SVM model is trained using the training data *X_train* and *y_train.*

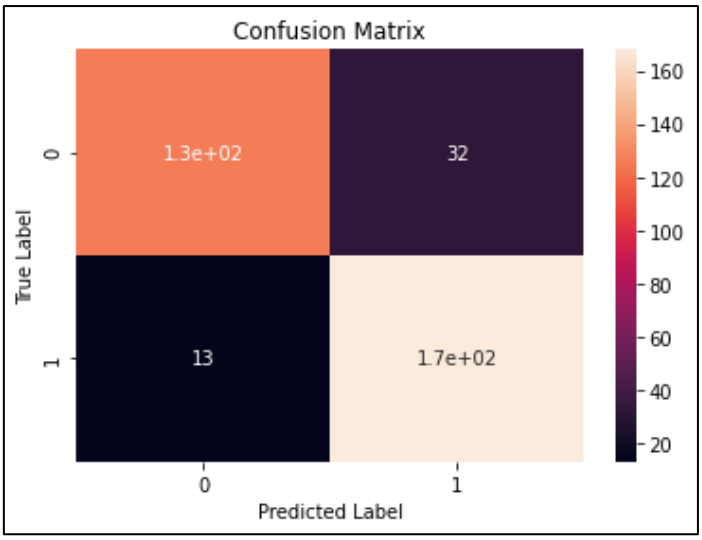
```
# Train the svm classifier
from sklearn.svm import SVC
clf = SVC(kernel='linear')
clf.fit(X_train, y_train)
```

```
SVC(kernel='linear')
```

**Predicting the Target Values:** The target values for the test data are predicted using the *clf.predict(X_test)* method.

```
# Predict the target values for the test data
y_pred = clf.predict(X_test)
```

**Evaluating the Model:** The accuracy of the model is calculated using the *accuracy_score* method from scikit-learn. The confusion matrix is plotted to check for the performance of the model.

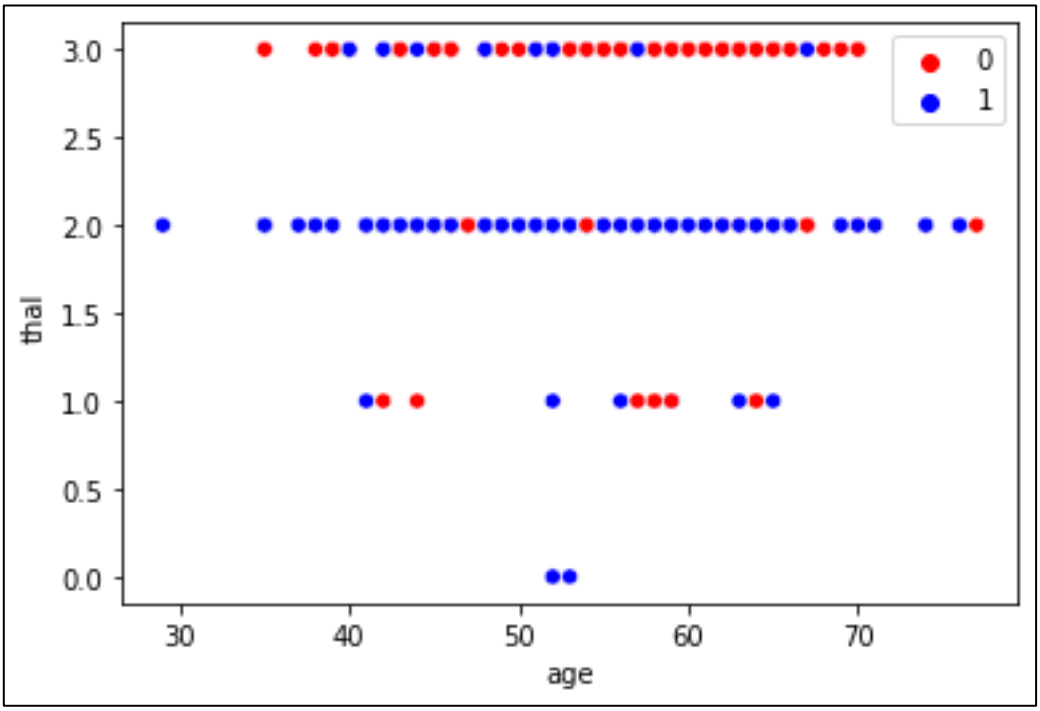


```
from sklearn.metrics import accuracy_score

# Calculate the accuracy of the classifier
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.8672566371681416
```

**Results:** The value of the accuracy score we get is 0.86 which means that our model is predicting 86% of the test data correctly which is a very fair number to prove that the model performs well.

**Plotting the SVM Output:** The output of the SVM model is plotted as a scatterplot with age and *thal* as the features and the predicted label as the hue.




**Identify ethical and professional responsibilities associated with developing machine learning project.**

---

When developing machine learning models, it is important to consider the ethical and professional responsibilities associated with the use of such models.

- **Explainability:** The predictions made by machine learning models should be understandable and transparent to those who use them.
- **Bias and Fairness:** It is important to ensure that the training data used to build the models is representative of the population and free from any biases that could lead to unfair or discriminatory predictions.
- **Responsibility:** The developers of machine learning models have a responsibility to ensure that their models are used for ethical and legal purposes.

By following these ethical and professional responsibilities, machine learning developers can help ensure that their models are used to improve the lives of individuals and communities, rather than causing harm.