|  |
| --- |
| HEART DISEASE PREDICTION |

|  |
| --- |
| Electrical & Computer Engineering & Computer Science (ECECS) |

|  |  |
| --- | --- |
| SPRING 23 |  |



Table of Contents

Abstract3

Data Collection4

Data Cleaning5

Data Understanding6

Data Preparation7

Methodology8

Decision Tree Classifier9

Random Forest10

Evaluation11

Deployment12

End Results13

Conclusion14

Contribution15

###### Heart Disease Prediction

Logo, company name

Description automatically generated

University of New Haven

Distributed and Scalable Data Engineering

DSCI-6007-03

Team Members

Sravan Kumar Reddy Pebbeti (Data Analyst)

Rakshala Mukka (Data Engineer)

Keerthana Krishnamoorthy (Machine Learning Engineer)

Svachuta Siva Sai Krishna Prasad Gollavilli (Lead Data Scientist)

###### Abstract

The objective of this project is to determine if a patient is suffering from heart disease by employing various detection techniques. To achieve this, the study utilized different models such as Logistic Regression, Random Forest, and Decision Tree Classifier, which were compared to obtain precise predictions of the patients' conditions. This intelligent system leverages artificial intelligence (AI) strategies to make the most accurate disease predictions based on patient information. Patients can then seek advice from medical experts based on the system's output. The study employed a dataset that categorized patients as having or not having heart disease based on specific characteristics. The aim was to use this data to develop a model that can predict whether a patient has the disease or not. The findings demonstrate that all four models can be utilized to detect heart diseases in patients; however, the study identifies the most effective model by comparing the accuracy.

###### Data Collection

Our data collection involved systematically gathering and measuring information on specific variables of interest, which enabled us to address important questions and make informed decisions about the future. We took great care to ensure that the data we obtained was both relevant and validated. To ensure the quality and validity of the data, we opted to use Kaggle as our source site.

Graphical user interface, application, table, Excel

Description automatically generated

###### Data Cleaning

In order to improve the quality of our information and make more accurate decisions, we carried out a process known as Data Cleaning on our raw data. Here are the steps we took to achieve a cleaner dataset:

1. We eliminated any duplicate values that arose from combining multiple datasets and removed irrelevant observations that were not relevant to the problem at hand.

2. We addressed missing values using a variety of techniques, such as Imputation, dropping features or observations with missing values.

3. We reformatted the data types, such as Boolean, numeric, and Datetime, as needed.

4. Finally, we restructured and validated the data by reformatting strings and ensuring data consistency.

###### Data Understanding

We have imported the data from the data set using different libraries like numpy, pandas, matplotlib and seaborn. After loading the data, we read the dataset. These are the parts of the data loading. Our dataset consists of fourteen columns which are

1. Age: patient’s age

2. Sex: Gender of the patient (Male = 1, Female = 0),

3. Chest pain type(CP): Includes pressure, fullness, burning or tightness in the chest .

4. Trestbps: A person’s resting blood pressure >120.

5. Chol: cholesterol levels. Good = less than 200, Moderate = 200-239, High = >240.

6. Fasting blood sugar(FBS): Measures the blood sugar after the overnight fast. 0 = normal, 1 = abnormal.

7. Resting electro cartographic results(rest ECG): Normal heart rate beats 60-100 per minute. 0 = normal, 1 = abnormal.

8. Thalach: Persons maximum heart rate achieved.

9. Exercise included angina(exang): The pain or stress that occurs after exercise. 0 = normal, 1 = abnormal.

10. Oldpeak: Exercise relative to rest. Records the slope of the peak exercise whether up, flat or down. Ranges between 0 to 5.

11. Slope: It is ST segment shift relative to the exercise-induced increments in patients heart rate.

12. CA: Coronary artery disease. Patients records of having CA. 0 no records, 1 or 2 had the cardiac attacks in the past.

13. Thal: A blood disorder called thalassemia.

14. Target.

By providing all the fields required we can predict whether the patient has the heart related diseases or not. If our end result is the integer valued 0 then that particular patient has no heart disease and if the result is integer 1 then the patient has heart disease. We used info, dtypes, shape, describe to understand our dataset.

###### Data Preparation

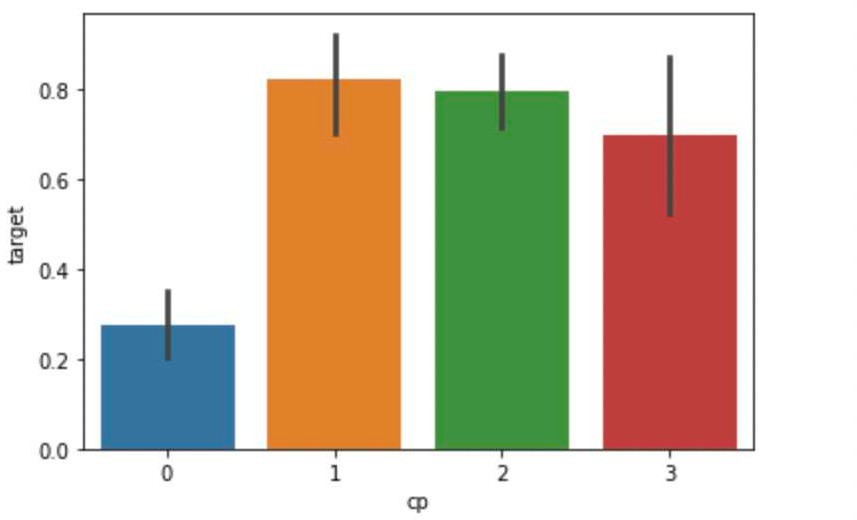
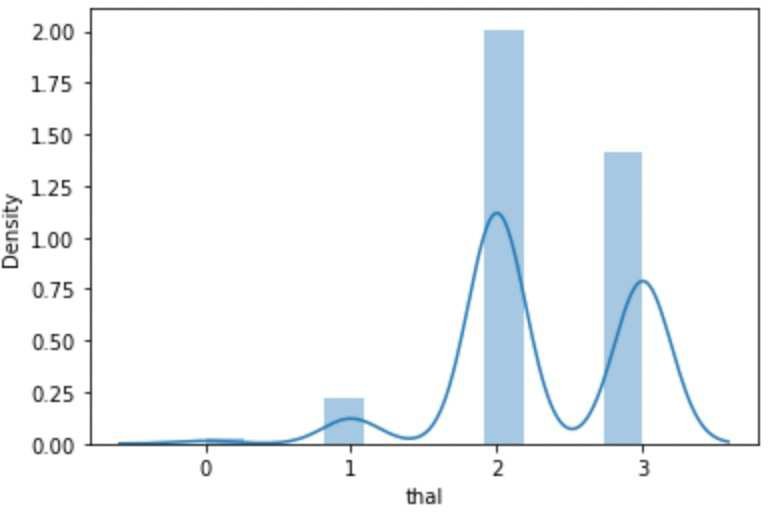
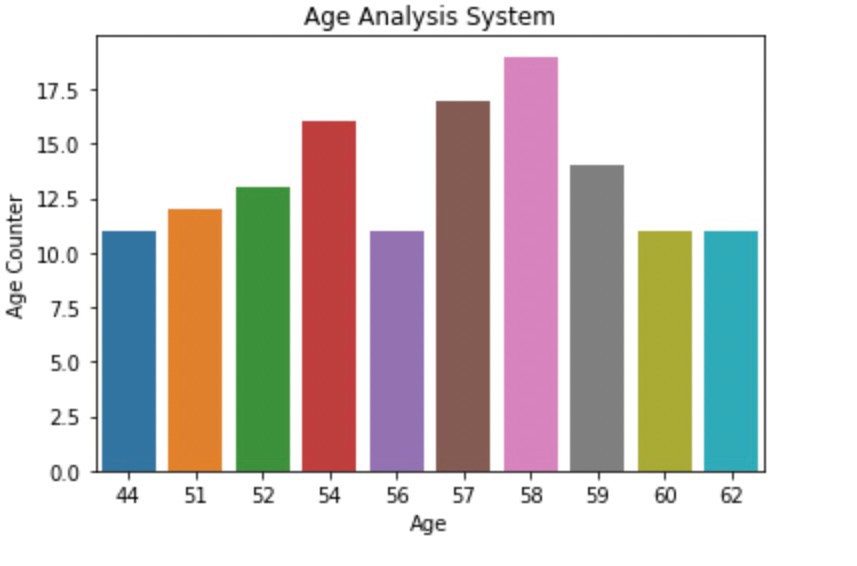
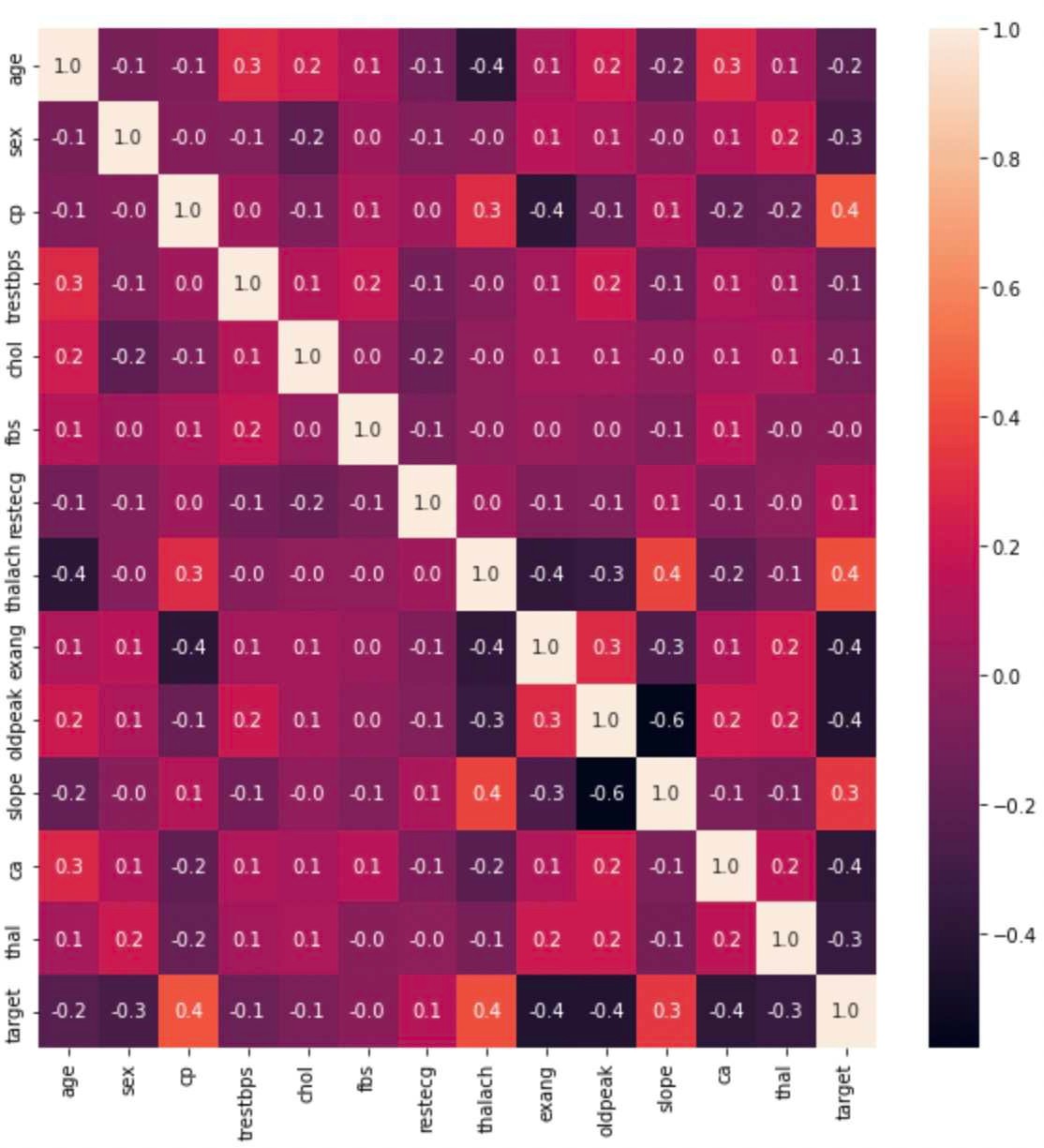
We conducted Exploratory Data Analysis (EDA) to identify any underlying patterns in the data, detect outliers, and test assumptions in order to find a model that best fits the data. EDA generally includes four main types of analysis, which are:

1. Univariate non-graphical analysis: This involves observing the population and analyzing the distribution of a single variable, such as measures of central tendency, measures of spread, and outlier detection.

2. Univariate graphical analysis: This involves using graphs to analyze a single variable, such as histograms, box plots, and stem-and-leaf plots.

3. Multivariate non-graphical analysis: This involves using techniques to analyze the relationship between two or more variables, such as covariance and correlation.

4. Multivariate graphical analysis: This involves using graphical methods to analyze the relationship between two or more variables, such as bar plots and scatterplots.



###### Methodology

In this project, we performed data modeling to develop the most effective approach for obtaining accurate results using three different algorithms: Random Forest Classification, Logistic Regression, and Decision Tree. We determined the best model for predicting heart disease in patients based on the highest accuracy rate achieved by the algorithms.

To evaluate the accuracy of our model, we utilized the train and test approach, which involved splitting the data into two sets: an 80% training set and a 20% testing set. We trained our model using the training set and then tested the model using the testing set. The accuracy rates of the algorithms were compared to determine which model was the most effective for predicting heart disease in patients.

Graphical user interface, text, application, email

Description automatically generated

We achieved an accuracy of 85.2% using Logistic regression classifier.

###### Decision Tree Classifier

The second classifier we used, decision trees, works by splitting up our dataset into smaller sections based on specific storing requirements. The number of samples in each subset is reduced with each dataset decision, creating a new subset. The network divides the data into groups that each include a single data point, and these examples are then classified according to a key that has been assigned.

We imported libraries, read the preprocessed dataset, and divided the data into training and testing as mentioned above. We have now determined whether the model fits the data. Then, in order to conduct an objective assessment and determine the model's accuracy score, we made predictions about the values using the classifier model.

The accuracy of the training and test sets was calculated, and we discovered that the results are consistent. We therefore verified that the model is not overfitting. We tested the classifier and determined the accuracy rate of the decision tree classifier before making predictions after ensuring that the model fits.

Graphical user interface, text, application, email

Description automatically generated

We achieved an accuracy of 81.9% using Decision Tree classifier.

###### Random Forest Classifier

The final classifier we employed is a random tree classifier, which combines many decision tree classifiers on different subsamples of the dataset and uses averaging to increase predicted accuracy and reduce over-fitting. As said before, we took into account importing libraries, our data set, and performing EDA. Create training and test sets from the data. For a binary probabilistic classifier, we now have the highest accuracy possible. To determine which classifier had the highest accuracy among the three, we compared the present accuracy of earlier classifiers with the maximum accuracy of the random forest classifier. The model was then checked to make sure it was correctly and without overfitting the dataset. Graphical user interface, text, application, email

Description automatically generated

We achieved an accuracy of 90.1% using Random Forest classifier.

###### Evaluation

In order to compare the accuracy scores of the three classifiers more effectively, we printed the findings and then displayed them using a bar plot.

Chart, bar chart

Description automatically generated

Therefore, we determined that the Random forest classifier model had the greatest accuracy score, 90.16%, and was therefore the best fitting model.

###### Deployment

This is the project's major stage for deploying and loading the model to the disc so that the outcomes may be compared. Additionally, can put the best model into practical use. The environment where we deploy the application will typically differ from the environment where we train it. Therefore, deployment enables us to ensure that our model persists naturally across all settings.

The pickle module, which serializes the object first before writing it into the file, was used to save the model to the disc. We utilized the load function of the Python pickle language to read the object's pickle byte stream.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generatedBased on our training data that we got from the model, we utilized the predict function to test the data, which returns the labels of the data supplied as an argument. The relevant output was written out at the conclusion.

Text

Description automatically generated

###### End Results

A screenshot of a computer

Description automatically generated with medium confidenceWe verified the final results in the local web page after deploying and saving our model. Our group split into two classifiers. Patients with a score of 1 have heart disease, whereas those with a score of 0 don't. The particular patient must fill out all fourteen fields on the website, which we have provided. The website will then convert the data back to source code and generate the cardiac conduction class based on the assumption that a user is present. the output we generated using the sample data offered in the earlier stage. The user receives the final result as an integer of 1 or 0, with 1 denoting the presence of cardiac disease and 0 denoting the absence of such disease.

A picture containing calendar

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

###### Conclusion

1. In this study, we were able to predict the likelihood that the patient would experience a specific type of heart disease by comparing the patient or user attributes. Based on variables such as age, blood pressure, fasting blood sugar, ECG resting results, thalamus, exercise-induced angina (exang), and more, this model helps us to compare a patient's characteristics and ascertain whether they are likely to acquire a disease.
2. The patient is a man, around the age of 21, with chest pain, a cholesterol level of 233, no resting ECG, and heart illness, according to the above-entered values for the patient. Therefore, patients with related issues may also be susceptible to comparable heart ailments. Medical facilities can use this to alert patients and protect them from future health crises.
3. Such models are capable of being used to simulate a wide range of medical disorders, including diabetes, cancer, heart disease, and mental health issues in addition to heart and cardiac conditions.

Contributions

* [https://www.kaggle.com/kralmachine/analyzing](http://www.kaggle.com/kralmachine/analyzing-the-heart-disease)-the-[heart](http://www.kaggle.com/kralmachine/analyzing-the-heart-disease)-[disease](http://www.kaggle.com/kralmachine/analyzing-the-heart-disease)
* https://towardsdatascience.com/simple-linear-regression-model-using-python- machine-learning-eab7924d18b4
* https://github.com/jupyterlab/jupyterlab
* [https://www.atmosera.com/blog/creating](http://www.atmosera.com/blog/creating-a-simple-linear-regression-machine-)-a-[simple](http://www.atmosera.com/blog/creating-a-simple-linear-regression-machine-)-li[near](http://www.atmosera.com/blog/creating-a-simple-linear-regression-machine-)-[regres](http://www.atmosera.com/blog/creating-a-simple-linear-regression-machine-)s[ion-machine-](http://www.atmosera.com/blog/creating-a-simple-linear-regression-machine-) learning-model-with-scikit-learn/
* [https://www.ibm.com/support/pages/producing](http://www.ibm.com/support/pages/producing-line-graph-dual-y-axes-spss-statistics)-line-[graph](http://www.ibm.com/support/pages/producing-line-graph-dual-y-axes-spss-statistics)-[dual](http://www.ibm.com/support/pages/producing-line-graph-dual-y-axes-spss-statistics)-[y-axes-spss-statistics](http://www.ibm.com/support/pages/producing-line-graph-dual-y-axes-spss-statistics)
* https://[www.open.ac.uk/socialsciences/spsstutorial/files/tutorials/graphs.pdf](http://www.open.ac.uk/socialsciences/spsstutorial/files/tutorials/graphs.pdf)
* https://jakevdp.github.io/PythonDataScienceHandbook