CYO Project report on Prediction of Heart Disease

Deeksha RV

3/4/2022

Contents

Abstract

The following report entails the contents of the second and final Data Science project that I finished, as part of the *HarvardX PH125.9x Data Science: Capstone* final course. For this project, I created **a machine-learning algorithm that predicts if a person will, or will not get Heart Disease, based on selected parameters.** The reason I selected this topic is because I wanted to work with some biological data. Moreover, according to 2019 worldwide statistics, 18.6 million people died of cardiovascular disease, globally, marking a 17.1% increase in the number of cases over the last decade; which makes this research-topic very challenging yet interesting, from the view of a data-scientist. A number of research papers have been published on this topic, and the most popular models used were K-Nearest-Neighbours, Naive Beyes and the Random Forest model.

The dataset was first downloaded as a csv file from Kaggle and checked for missing values. A total of 11 predictors were used to build the algorithm. The data was initially explored and analysed to understand the trends in the variables. Various visualization tools such as boxplots, histograms and bar graphs were drawn to extract useful insights from the data. Spearman's correlation method was used to find the measure of correlation of the predictors with the response variable. Various models such as Linear Discriminant Analysis (LDA), Loess, Quadratic Discriminant Analysis (QDA), Random Forest etc were fitted, and their accuracies were evaluated. A final ensemble model was created to check the final accuracy of the algorithm. The various algorithms built, were compared to find the one that gave the best accuracy.

Introduction

A prediction algorithm is a series of building blocks that are assembled together to make a prediction, or a forecast for an unknown event. Often, prediction algorithms are created to understand how an unknown variable changes with respect to certain known variables, related to the former. This is where the role of Regression steps in. Regression is a powerful statistical tool that helps understand the trends in data and how each variables correlates to the other. This makes predictions easier and more reliable. The most common form of regression used to make prediction algorithms is "Linear Regression". Although it is the easiest tool available, it has its own limitations; linear regression assumes that the the dependant and independant variables vary linearly, and it is sensitive to outliers, which is not always a good thing. Hence the algorithm built in this project was created beyond Linear Regression, to get a better accuracy. The following machine-learining models were used for this project:

1. *Linear Discriminant Analysis* - LDA is typically used when the response variable contains 2 or more classes. It is mainly preferred when the sample size is small, so we don't necessarily have to supply large amounts of data for LDA to make a prediction, making it very handy. Another important point to note is that, LDA assumes that the input data is *normally distributed.* Hence it is always the best idea to normalize/scale our data before using this model.

2. *Quadratic Discriminant Analysis* - QDA is a variation of LDA, mainly differing such that it separates data in a non-linear manner unlike LDA. The main drawback of this model is that, unlike LDA, it cannot be used as a technique to reduce the dimensions of our data.

3. *Logistic Regression* - Just like Linear Regression, Logistic regression is a useful statisitical tool to predict binary outcomes. It works best when there are no outliers present in the data. Moroever, the output provided by Logistic regression is discrete in nature.

4. *Loess* - expanded as Locally Weighted Scatterplot Smoothing, the Loess model is yet another tool used in regression analysis for understanding the relationship between variables. Scatterplots are sometimes difficult to interpret due to the presence of some noisy data points and outliers. A loess model helps smooth out data points by fitting a line to the scatterplot and hence shows a clear trend in the data.

5. *K nearest neighbours* - The KNN model is a distance-based model that computes the distance between the observations of the features, and uses that to relate our dependant and independant variables. The value of "k" selected, controls the accuracy of our estimate; larger values of "k" result in smooth estimates, whereas smaller values of "k" result in flexible estimates. The value of "k" is user-defined.

6. *Random Forest* - Random Forest model is an improved version of decision trees. The goal of a random forest model is to improve the prediction accuracy by considering the average of multiple decision trees. It has certain parameters which can be tuned by the user, in order to get better accuracy.

Viewing and Understanding the Dataset

The dataset used for this project was directly obtained from kaggle, and was created by combining five independant heart datasets, whose data was procured from various hospitals.
First, the dataset was imported from the Kaggle website.

Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0
39	M	NAP	120	339	0	Normal	170	N	0.0	Up	0

The dataset consists of 918 rows and 12 columns. Out of 12 columns, 11 predictors namely Age, Sex, Chest Pain Type, Resting BP, Cholesterol, Fasting Blood Sugar, Resting ECG, Maximum Heart Rate, Exercise Angina, Old Peak and ST Slope, have been used to predict the response variable, HeartDisease. The HeartDisease variable has a binary outcome; 1 indicating the presence of heart disease and 0 indicating its absence. A detailed description of each predictor is given below:

1. Age - Age of a person in years

2. Sex - M for male, F for female

3. ChestPainType -
   ASY = Asymptomatic
   ATA = Atypical Angina
   NAP = Non Anginal Pain
   TA = Typical Angina

4. RestingBP - Systolic, in mm Hg, at time of admission in hospital

5. Cholesterol - Serum cholestoral in mg/dl

6. FastingBS - Fasting Blood Sugar $>$ 120 mg/dl, (1 = true; 0 = false)

7. Resting ECG - Resting Electro-Cardiographic results
   $Normal$ = normal
   ST = having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of $>$ 0.05 mV)
   LVH = showing probable or definite left ventricular hypertrophy by Estes' criteria

8. MaxHR - Maximum Heart Rate achieved

9. ExerciseAngina - Exercise induced Angina
   Y = yes
   N = no

10. Oldpeak - ST depression induced by exercise relative to rest

11. STSlope - the slope of the peak exercise ST segment
    Up = upsloping
    $Flat$ = flat
    $Down$ = downsloping

There are no missing values in this dataset.

Modifying the Dataset

The following changes were made to the existing dataset, to make it easier for analysis:

First, the categories of ExerciseAngina were changed from *Yes* and *No* to *1* and *0* respectively:

```
heart_attack$ExerciseAngina <- ifelse(heart_attack$ExerciseAngina == "Y",1,0)
```

Next, the categories of Sex variable were changed from *M* and *F* to *1* and *0* respectively:

```
heart_attack$Sex <- ifelse(heart_attack$Sex == "M",1,0)
```

The categories of ChestPainType variable were also changed as follows:

```
heart_attack$ChestPainType <- as.factor(heart_attack$ChestPainType)
heart_attack$ChestPainType <- as.numeric(heart_attack$ChestPainType)
```

RestingECG values were made numeric:

```
heart_attack$RestingECG <- as.numeric(as.factor(heart_attack$RestingECG))
```

STSlope values were similarly changed:

```
heart_attack$ST_Slope <- as.numeric(as.factor(heart_attack$ST_Slope))
```

The modified dataset looks like this:

Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
40	1	2	140	289	0	2	172	0	0.0	3	0
49	0	3	160	180	0	2	156	0	1.0	2	1
37	1	2	130	283	0	3	98	0	0.0	3	0
48	0	1	138	214	0	2	108	1	1.5	2	1
54	1	3	150	195	0	2	122	0	0.0	3	0
39	1	3	120	339	0	2	170	0	0.0	3	0

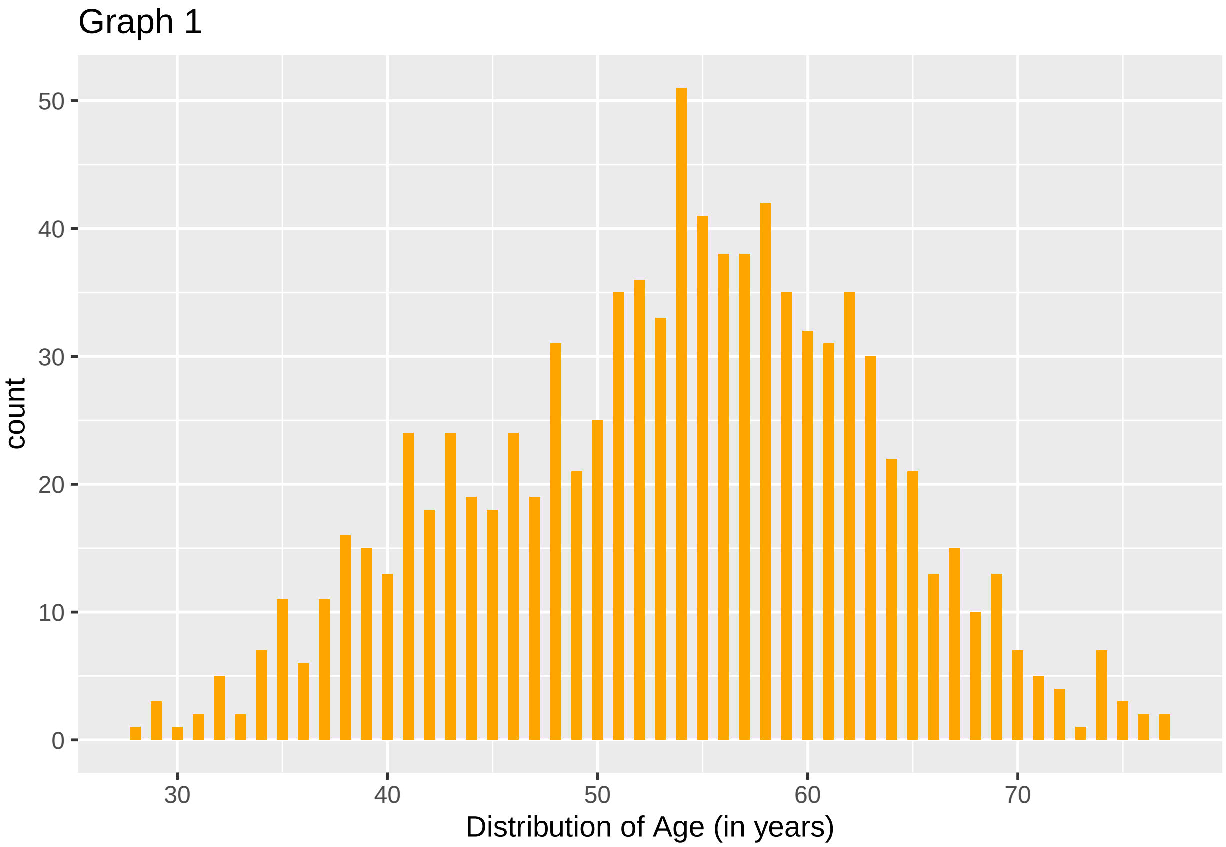
Exploratory Data Analysis

Going further into the data, we find that the dataset can be split by gender into 725 men and 193 women. Also, 55.3% of the population in the dataset have a definite chance of getting heart disease.

The data can be summarized in total, as follows:

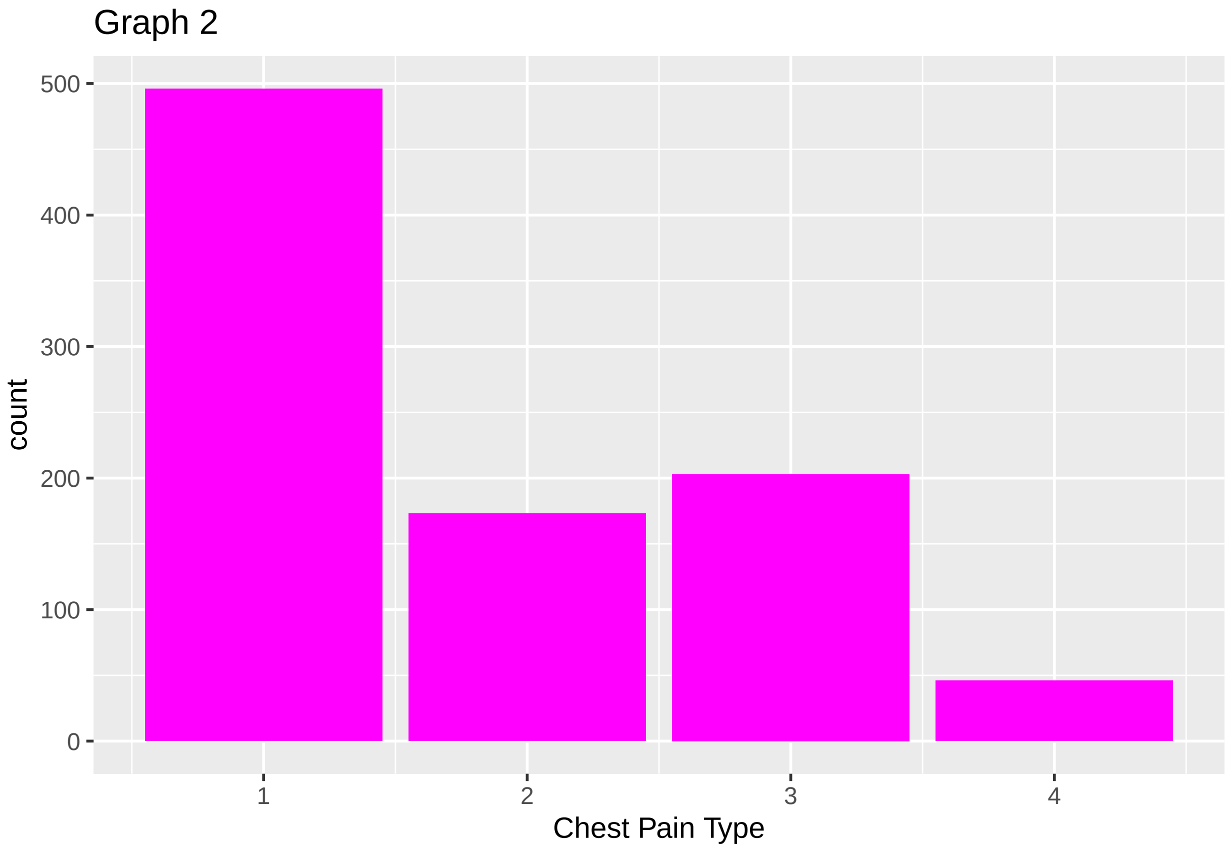
	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
	Min. :28.00	Min. :0.0000	Min. :1.000	Min. : 0.0	Min. : 0.0	Min. :0.0000	Min. :1.000	Min. : 60.0	Min. :0.0000	Min. :-2.6000	Min. :1.000	Min. :0.0000
	1st Qu.:47.00	1st Qu.:1.0000	1st Qu.:1.000	1st Qu.:120.0	1st Qu.:173.2	1st Qu.:0.0000	1st Qu.:2.000	1st Qu.:120.0	1st Qu.:0.0000	1st Qu.: 0.0000	1st Qu.:2.000	1st Qu.:0.0000
	Median :54.00	Median :1.0000	Median :1.000	Median :130.0	Median :223.0	Median :0.0000	Median :2.000	Median :138.0	Median :0.0000	Median : 0.6000	Median :2.000	Median :1.0000
	Mean :53.51	Mean :0.7898	Mean :1.781	Mean :132.4	Mean :198.8	Mean :0.2331	Mean :1.989	Mean :136.8	Mean :0.4041	Mean : 0.8874	Mean :2.362	Mean :0.5534
	3rd Qu.:60.00	3rd Qu.:1.0000	3rd Qu.:3.000	3rd Qu.:140.0	3rd Qu.:267.0	3rd Qu.:0.0000	3rd Qu.:2.000	3rd Qu.:156.0	3rd Qu.:1.0000	3rd Qu.: 1.5000	3rd Qu.:3.000	3rd Qu.:1.0000
	Max. :77.00	Max. :1.0000	Max. :4.000	Max. :200.0	Max. :603.0	Max. :1.0000	Max. :3.000	Max. :202.0	Max. :1.0000	Max. : 6.2000	Max. :3.000	Max. :1.0000

Looking into some variables, the distribution of Age can be visualized below as a histogram:



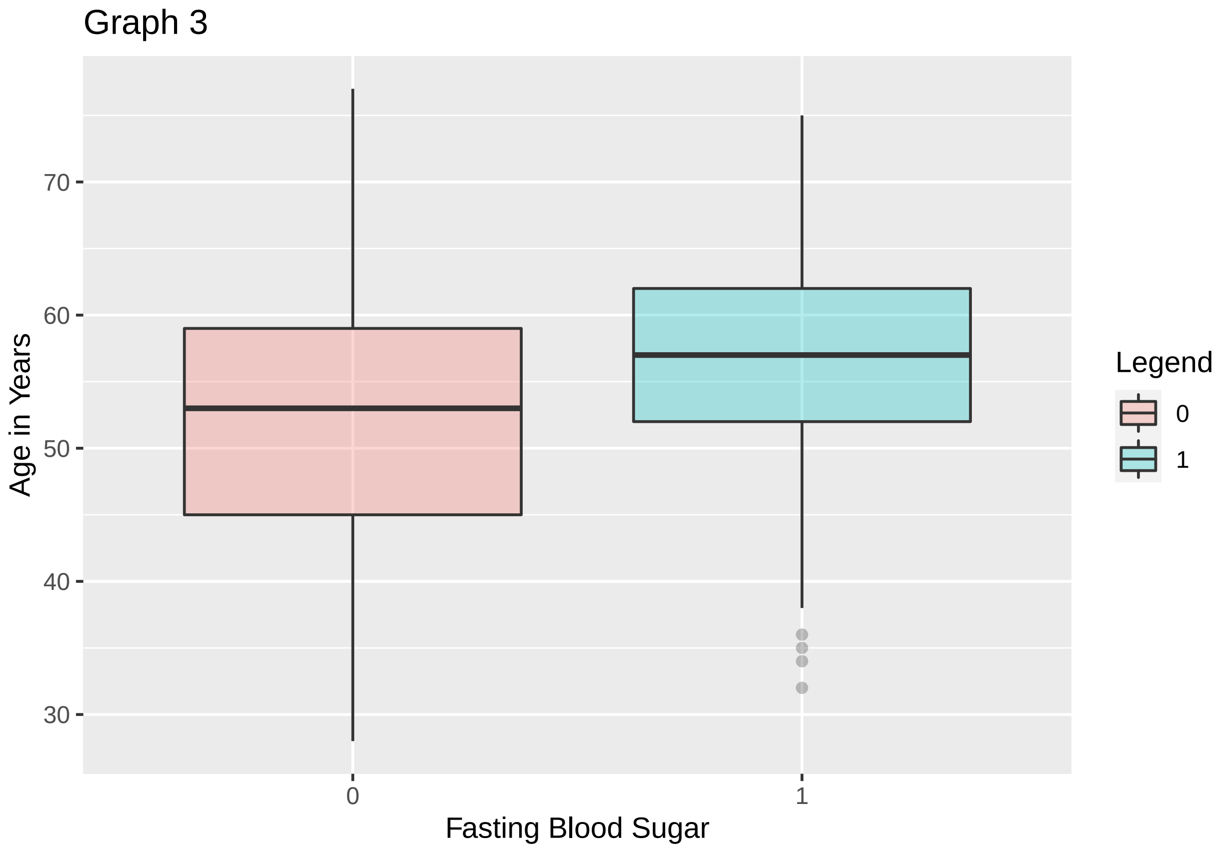
The graph is roughly bell-shaped, implying that the data follows an approximately normal distrubution. The majority of the population is aged between *50-60 years*. The least age of a person is *28 years*, and the oldest person is *77 years* old. The mean age of a person from this dataset is *53.5 years*, and the median age is *54 years*.

The types of Chest Pain is visualized as follows:



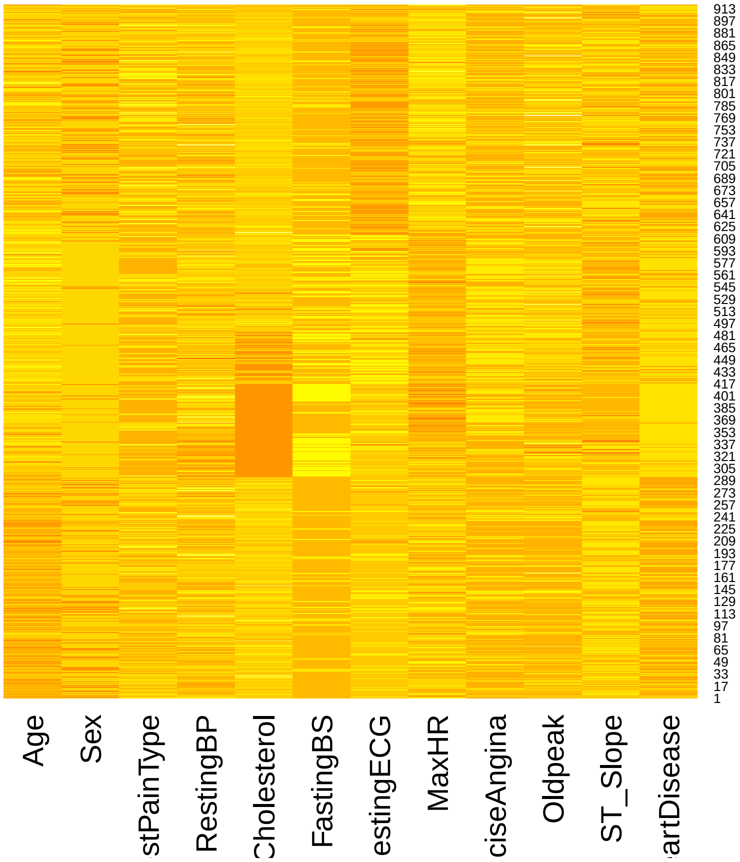
As it can be seen, the majority proportion of the population, 496 people, has *Asymptomatic Chest Pain* (corresponding to a value of *1*), followed by 203 people who have *Non-Anginal pain* (=*3*). The least proportion of population, containing 46 patients, has *Typical Anginal Pain* (=*4*).

Let us now see a bi-variate analysis. This is how Fasting Blood Sugar varies with Age:



We can see that some outliers are present for the values of Fasting Blood Sugar = *1*. As given before, a value of *1* for Fasting Blood Sugar indicates a blood sugar level greater than 120 mg/dl. People aged between 52-60 years, seem to have a higher level of Fasting Blood Sugar, the average age being 56 years. Higher levels of this predictor are more likely to cause heart disease.

Let us now see a spatial heatmap of all our features to understand the overall trends in our dataset:



The darker oranges indicate higher magnitude of values for each variable. The ochres and yellows indicate lesser magnitude for the same. For example, taking the predictor Cholesterol, the first 300 rows show light yellow cells, indicating lower cholesterol levels for these people. Whereas, for the rows 300-500, the color changes to orange, indicating higher cholesterol levels for these people.

We will now use Spearman's correlation method to find the correlation of all predictors with our HeartDisease variable:

Parameter	correlation with heart disease
Age	0.2895757
Sex	0.3054449
ChestPainType	-0.4425160
Resting BP	0.1138663
Cholesterol	-0.1398731
Fasting BS	0.2672912
Resting ECG	0.0568940
Max HR	-0.4048268
Exercise Angina	0.4942820
Old peak	0.4190461
ST Slope	-0.5919128

As observed, while some predictors such as Age, Sex and ExerciseAngina show a high *positive* correlation, some other predictors such as ChestPainType, Cholesterol and Max HR show *negative* correlation. The highest positive correlation for HeartDisease is found with ExerciseAngina.

Building the Algorithm

Before we start fitting models, we will have to prepare our dataset. For this, we will start by creating a vector, *"y"*, which will hold the binary outcomes under the HeartDisease variable.

```
y <- heart_attack$HeartDisease
```

We will now subset our dataset to include only the values of the predictors.

Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope
40	1	2	140	289	0	2	172	0	0.0	3
49	0	3	160	180	0	2	156	0	1.0	2
37	1	2	130	283	0	3	98	0	0.0	3
48	0	1	138	214	0	2	108	1	1.5	2
54	1	3	150	195	0	2	122	0	0.0	3
39	1	3	120	339	0	2	170	0	0.0	3

Scaling our dataset is necessary for our models to give better accuaracy. Scaling changes the range of our data values and ensures that the variance of all the predictors is equal to 1; that is, all our predictors change trends equally. There are different functions that help scale a dataset. The scaled dataset looks like this:

Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope
-1.4323590	0.5156713	0.2289073	0.4106850	0.8246208	-0.551041	0.0172451	1.3821748	-0.8231076	-0.8319789	1.0515406
-0.4782229	-1.9371073	1.2743644	1.4909396	-0.1718674	-0.551041	0.0172451	0.7537463	-0.8231076	0.1056060	-0.5957534
-1.7504044	0.5156713	0.2289073	-0.1294423	0.7697682	-0.551041	1.6003466	-1.5243071	-0.8231076	-0.8319789	1.0515406
-0.5842380	-1.9371073	-0.8165498	0.3026596	0.1389638	-0.551041	0.0172451	-1.1315393	1.2135845	0.5743984	-0.5957534
0.0518527	0.5156713	1.2743644	0.9508123	-0.0347360	-0.551041	0.0172451	-0.5816643	-0.8231076	-0.8319789	1.0515406
-1.5383741	0.5156713	1.2743644	-0.6695696	1.2817254	-0.551041	0.0172451	1.3036212	-0.8231076	-0.8319789	1.0515406

We can now create the *test* and *train* sets, by splitting the data into 75% for training and 25% for testing. Selecting this ratio ensures that we neither overtrain nor undertrain our models.

```
# create test index
test_index <- createDataPartition(y, times = 1, p = 0.25, list = FALSE)
# test set of heart_attack
test_HA <- scaled_heart_attack[test_index,]

# test set of y
test_y <- y[test_index]

# train set of heart_attack
train_HA <- scaled_heart_attack[-test_index,]

# train set of y
train_y <- y[-test_index]
```

The vectors containing our outcomes (y, train_y, test_y) are converted into factors.

Now we are ready to fit our models. We will start with the LDA model:

Model	Accuracy
LDA	0.8565217

This model gives us an accuracy of 0.8565217, or 85.65%. The reason the accuracy is not good enough, is because the LDA model generally works best when our sample size is small; about 200 observations or lesser. As we are working with a larger dataset, the accuracy is naturally low.

We will now try a QDA model:

Model	Accuracy
LDA	0.8565217
QDA	0.8304348

Here, our QDA model gives an accuracy of 0.8304348, or 83.04%, which is even worse than our LDA model. We will try to improve the accuracy by using better models.

Let us try a Logistic Regression model now:

Model	Accuracy
LDA	0.8565217
QDA	0.8304348
Logistic Regression	0.8652174

The accuracy has now jumped up to 0.8652174, or 86.5%. We have made some progress from our QDA model.

We will also try a LOESS model:

Model	Accuracy
LDA	0.8565217
QDA	0.8304348
Logistic Regression	0.8652174
LOESS	0.8695652

Our accuracy has marginally improved now, to 0.8695652, or approximately 87%.

Let us now try a KNN model.A KNN model is a powerful model which can be adjusted to include user-defined values of k. The selection of k value is done from a pre-defined range that we provide to R. This range can only be selected by trial-and-error analysis. For our project, the following range of k values - odd numbers between 21 and 41 gave the highest accuracy:

Model	Accuracy
LDA	0.8565217
QDA	0.8304348
Logistic Regression	0.8652174
LOESS	0.8695652
KNN	0.8869565

The best k value for this model was found to be 23, which yielded a high accuracy of 0.8869565, or 88.7%. This is the best accuracy we have got so far, from all our models.

We will also try a Random Forest model. Just like the KNN model, the Random Forest model also can be tuned according to our requirement. We can change our *"mtry"* values using trial-and-error analysis to see which values yield the highest accuracy:

Model	Accuracy
LDA	0.8565217
QDA	0.8304348
Logistic Regression	0.8652174
LOESS	0.8695652
KNN	0.8869565
Random Forest	0.8826087

The best accuracy which we got for this model is 0.8826087 or 88.3%. It marginally lags behind our KNN model but makes the second-best prediction out of all our models so far.

We will now create an ensemble model by taking into account, the predictions of all the models we have created, and see how this model performs:

Model	Accuracy
LDA	0.8565217
QDA	0.8304348
Logistic Regression	0.8652174
LOESS	0.8695652
KNN	0.8869565
Random Forest	0.8826087
Ensemble	0.8695652

Our ensemble model has yielded an accuracy of 0.8695652 or 87% approximately.

Now, the accuracy of our ensemble model is not satisfactory enough, so we will create a second ensemble model, combining the predictions of the two of our best models; the KNN model and the Random Forest model, and see how well this combination can predict HeartDisease:

Model	Accuracy
LDA	0.8565217
QDA	0.8304348
Logistic Regression	0.8652174
LOESS	0.8695652
KNN	0.8869565
Random Forest	0.8826087
Ensemble	0.8695652
Final Ensemble	0.9000000

This model yields an accuracy of 0.9 or 90% exactly. The combination of KNN and Random Forest was able to make a much better prediction than the individual models themselves. Hence, this model will be taken as our final model for our algorithm.

Results and Conclusion

The final algorithm created in this project is the “Final Ensemble” model, which is a combination of the KNN and Random Forest Models. This model is able to make a prediction that yielded an accuracy of 90%, which is the best accuracy we got from all the models. It is important to note here that, the accuracies of the models keep changing every time the code is run, due to being sampled repeatedly. The accuracies recorded during this project were obtained by running each model twice. From this project, it has been understood that models such as QDA and LDA can only be used as baseline models, or if sample size is quite small, but it cannot be entirely relied upon for the final accuracy. In conclusion, of all the individual models used, the KNN model has proved to be the best model to make predictions. This is attributed to the flexibility of tuning the model.

Scope for Improvement

The current final model can be improved in the future, by fitting more number of prediction models into our data. Moreover, models such as KNN and Random Forest can also be tuned with different parameters to further improve their accuracy. With Random Forest, the number of trees may be increased for higher accuracy. Ensemble models of different combinations may be created and further comparison of accuracies can be made. Data mining techniques such as clustering, genetic algorithm and time series can be incorporated, which are beyond the scope of this project.

References

[1] https://newsroom.heart.org/news/heart-disease
[2] https://sciencing.com/disadvantages-linear-regression-8562780.html
[3] https://www.statology.org/linear-discriminant-analysis-in-r/#:~:text=Linear%20discriminant%20analysis%20is%20a,into
[4] https://www.statology.org/linear-discriminant-analysis/
[5] https://www.analyticsvidhya.com/blog/2020/12/beginners-take-how-logistic-regression-is-related-to-linear-regression/#:~:text=Linear%20Regression%20is%20used%20to,Logistic%20regression%20provides%20discreet%20outp
[6] https://www.statisticshowto.com/lowess-smoothing/
[7] https://bookdown.org/yihui/rmarkdown-cookbook/font-color.html
[8] https://learning.edx.org/course/course-v1:HarvardX+PH125.8x+1T2021/home