Heart Disease Prediction using Logistic Regression and K-Nearest Neighbours

As a part of the course on Data Science we are to choose our own project and use models / algorithms and predict the outcomes based on the dataset. The project, "Heart Disease Prediction using Logistic Regression and K-Nearest Neighbours" has been chosen by me. The project requires coding in R. The project report has been covered under four major heads, viz:

1. Introduction
2. Analysis
3. Results
4. Conclusion

1. Introduction

This section describes the dataset and variables, and summarizes the goal of the project and key steps that were performed.

1.1. Data Source

Context

This data set dates from 1988 and consists of four databases: Cleveland, Hungary, Switzerland, and Long Beach V. It contains 76 attributes, including the predicted attribute, but all published experiments refer to using a subset of 14 of them. The "target" field refers to the presence of heart disease in the patient. It is integer valued 0 = no disease and 1 = disease. The dataset utilized in this project comprises comprehensive information pertaining to patients diagnosed with heart disease. It encompasses a wide array of data derived from multiple factors. Should you wish to access and acquire this dataset, it is readily available for download on Kaggle. (https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset)

Here are some brief explanations of the variables :

- `age` : Age of the individual in years (integer)
- `sex` : Gender of the individual (1 = male; 0 = female) (integer)
- `cp` : Chest pain type (1 = typical angina; 2 = atypical angina; 3 = non-anginal pain; 4 = asymptomatic) (integer)
- `trestbps` : Resting blood pressure (in mm Hg on admission to the hospital) (integer)
- `chol` : Serum cholesterol in mg/dl (integer)
- `fbs` : Fasting blood sugar level > 120 mg/dl (1 = true; 0 = false) (integer)

- restecg : Resting electrocardiographic results (0 = normal; 1 = having ST:T; 2 = hypertrophy) (integer)
- thalach : Maximum heart rate achieved (integer)
- exang : Exercise induced angina (1 = yes; 0 = no) (integer)
- oldpeak : ST depression induced by exercise relative to rest (numeric)
- slope : The slope of the peak exercise ST segment (1 = upsloping; 2 = flat; 3 = downsloping) (integer)
- ca : Number of major vessels (0:3) colored by flourosopy (integer)
- thal : Thalassemia is an inherited blood disorder that affects the body's ability to produce haemoglobin and red blood cells. 0 = normal; 1 = fixed defect; 2 = reversable defect (integer)
- target : the predicted attribute : diagnosis of heart disease (angiographic disease status). Value 0 = no disease < 50% diameter narrowing; - Value 1 = disease > 50% diameter narrowing. (integer)

1.2. Objective

The main objective of this project is to predict whether a given person is at risk of having heart disease or not. This prediction is made by analysing several contributing factors, including age, cholesterol level, and the type of chest pain experienced by the individual etc.

In order to achieve this objective, we employ the following algorithms:

Logistic Regression: This algorithm is a statistical method that is commonly used for binary classification tasks, such as determining the presence or absence of heart disease. It calculates the probability of an individual having heart disease based on the input factors and then classifies them accordingly.

K-Nearest Neighbours (K-NN): K-Nearest Neighbours is a machine learning algorithm used for classification tasks. In this context, it works by identifying the k-nearest individuals in the dataset who share similar characteristics with the person being evaluated. The algorithm then predicts the presence of heart disease based on the majority class of these nearest neighbours.

1.3. Key Steps

1.3.1. Library

In this project, we have incorporated several essential libraries to facilitate our data analysis and model building. These libraries include:

```
> library(readr)

> library(dplyr)

> library(ggplot2)

> library(lmtest)

> library(caret)

> library(car)

> library(MLmetrics)

> library(class)

> library(GGally)
```

1.3.2. Data Preparation

The Data Preparation process consists of two primary activities: Loading Data and Data Wrangling. First, we Load Data by utilizing the readr library to import the raw dataset. Following this, we move to Data Wrangling, where we perform data cleaning, transformation, and structuring tasks, ensuring that the data is in an optimal and reliable state for further analysis and model development.

i. Load Data

The data downloaded from (https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset) to the local hard disk will be used for the purpose of loading of data. We Load Data by utilising the `readr` library to import the raw dataset.

```
> heart <- read.csv("/Users/georgemathew/Desktop/Desktop Folders/HP BACKUP/C DRIVE/Desktop/Kerala Summons/GEM/edX courses SBI/Data Science HarvardX/choose_your_own_project/Heart Disease_log_regrr_N_KNN/heart.csv")

> heart %>% head()
  age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
1  52   1  0      125  212   0       1     168     0     1.0     2  2    3
2  53   1  0      140  203   1       0     155     1     3.1     0  0    3
```

```
3   70    1    0     145     174     0      1        125        1       2.6      0    0    3
4   61    1    0     148     203     0      1        161        0       0.0      2    1    3
5   62    0    0     138     294     1      1        106        0       1.9      1    3    2
6   58    0    0     100     248     0      0        122        0       1.0      1    0    2
  target
1      0
2      0
3      0
4      0
5      0
6      1
```

ii. Data Wrangling

Data Wrangling is a pivotal phase in our project, where we focus on transforming and structuring the data to prepare it for analysis and modelling. To kick off this process, we begin by utilizing the `glimpse` function. This function is a handy feature provided by the `dplyr` library and allows us to swiftly gain insight into the data's structure. It displays a concise summary of the dataset, revealing information about the number of observations and variables, as well as the data types of each variable.

```
> str(heart)
'data.frame':    1025 obs. of  14 variables:
 $ age     : int  52 53 70 61 62 58 58 55 46 54 ...
 $ sex     : int  1 1 1 1 0 0 1 1 1 1 ...
 $ cp      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ trestbps: int  125 140 145 148 138 100 114 160 120 122 ...
 $ chol    : int  212 203 174 203 294 248 318 289 249 286 ...
```

$ fbs : int 0 1 0 0 1 0 0 0 0 0 ...

$ restecg : int 1 0 1 1 1 0 2 0 0 0 ...

$ thalach : int 168 155 125 161 106 122 140 145 144 116 ...

$ exang : int 0 1 1 0 0 0 0 1 0 1 ...

$ oldpeak : num 1 3.1 2.6 0 1.9 1 4.4 0.8 0.8 3.2 ...

$ slope : int 2 0 0 2 1 1 0 1 2 1 ...

$ ca : int 2 0 0 1 3 0 3 1 0 2 ...

$ thal : int 3 3 3 3 2 2 1 3 3 2 ...

$ target : int 0 0 0 0 0 1 0 0 0 0 ...

In the data examination process, we have identified several variables that need to be transformed into factor data types. These variables include sex, cp, fbs, restecg, exang, slope, ca, thal , and target. Factor data types allow us to more efficiently manage these categorical variables in our analysis and modelling, and also facilitate the creation of informative visualizations.

> heart <- heart %>% mutate(sex = as.factor(sex), cp = as.factor(cp), fbs = as.factor(fbs), restecg = as.factor(restecg), exang = as.factor(exang), slope = as.factor(slope), ca = as.factor(ca), thal = as.factor(thal), target = as.factor(target))

> str(heart)

'data.frame': 1025 obs. of 14 variables:

$ age : int 52 53 70 61 62 58 58 55 46 54 ...

$ sex : Factor w/ 2 levels "0","1": 2 2 2 2 1 1 2 2 2 2 ...

$ cp : Factor w/ 4 levels "0","1","2","3": 1 1 1 1 1 1 1 1 1 1 ...

$ trestbps: int 125 140 145 148 138 100 114 160 120 122 ...

$ chol : int 212 203 174 203 294 248 318 289 249 286 ...

$ fbs : Factor w/ 2 levels "0","1": 1 2 1 1 2 1 1 1 1 1 ...

$ restecg : Factor w/ 3 levels "0","1","2": 2 1 2 2 2 1 3 1 1 1 ...

```
$ thalach : int  168 155 125 161 106 122 140 145 144 116 ...

$ exang   : Factor w/ 2 levels "0","1": 1 2 2 1 1 1 1 2 1 2 ...

$ oldpeak : num  1 3.1 2.6 0 1.9 1 4.4 0.8 0.8 3.2 ...

$ slope   : Factor w/ 3 levels "0","1","2": 3 1 1 3 2 2 1 2 3 2 ...

$ ca      : Factor w/ 5 levels "0","1","2","3",..: 3 1 1 2 4 1 4 2 1 3 ...

$ thal    : Factor w/ 4 levels "0","1","2","3": 4 4 4 4 3 3 2 4 4 3 ...

$ target  : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
```

2. Analysis

2.1. Exploratory Data Analysis

The Exploratory Data Analysis (EDA) section of our project plays a pivotal role in unveiling crucial insights from the dataset.

i. Proportions Target Class

Assess the distribution of target variable to understand the prevalence of heart disease.

```
> prop.table(table(heart$target))

        0         1

0.4868293 0.5131707
```

The distribution of the target class in dataset demonstrates a state of balance. With approximately 48.68% of instances falling into category 0 and around 51.32% into category 1, there is no significant imbalance between the two classes. A balanced distribution of the target class is a favourable characteristic for our analysis and modelling processes. It ensures that both outcomes are adequately represented in dataset, thereby contributing to the robustness and reliability of predictive models.

ii. Missing Value

Which involves an examination of missing data points, ensuring data completeness and integrity.

```
> anyNA(heart)

[1] FALSE

> colSums(is.na(heart))

    age      sex       cp trestbps     chol      fbs  restecg  thalach

      0        0        0        0        0        0        0        0

  exang  oldpeak    slope       ca     thal   target

      0        0        0        0        0        0
```

The dataset has no missing values, signifying a high level of data completeness and integrity. This absence of gaps in the data streamlines analysis and modelling processes, instilling confidence in the reliability and efficiency of endeavours.

iii. <u>Correlation</u>

We delve into the relationships between variables to uncover potential patterns and associations.

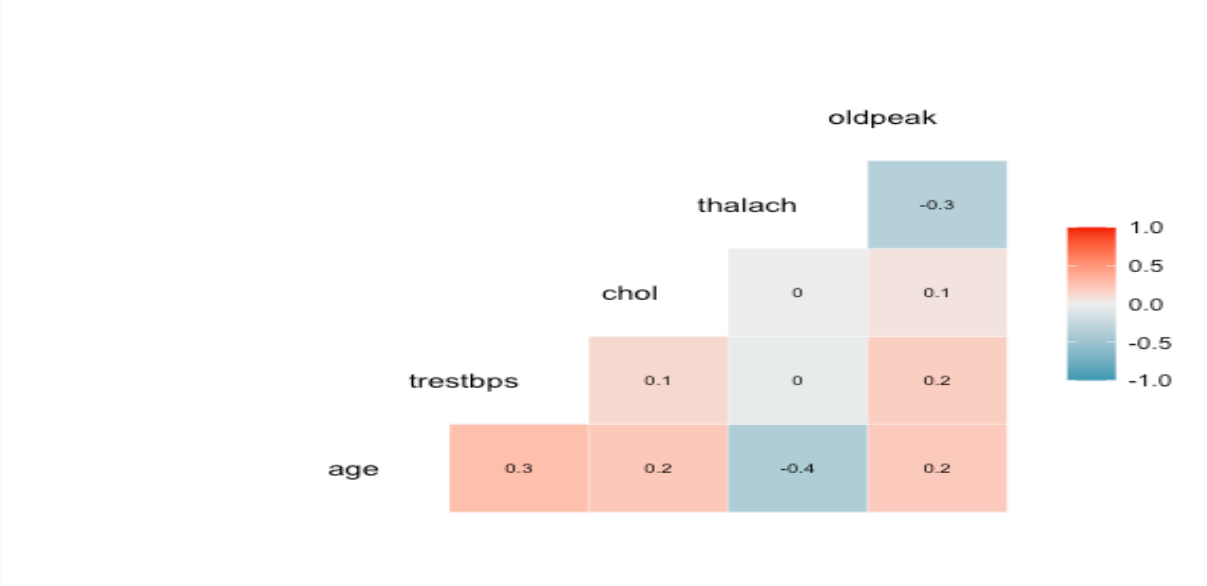
```
> a > ggcorr(heart, label = TRUE, label_size = 2.5, hjust = 1, layout.exp = 2)

> plot(a)

Warning message:

In ggcorr(heart, label = TRUE, label_size = 2.5, hjust = 1, layout.exp = 2) :

data in column(s) 'sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal', 'target' are not numeric and were ignored
```



From the correlation checking process, it's evident that only some correlations between variables can be displayed because the ggcorr library ignores data types other than numeric. Among the displayed correlations, there is a negative correlation between oldpeak and thalach, with a value of -0.3. A negative correlation indicates that when one variable increases, the other tends to decrease, and vice versa. In this context, the value of -0.3 suggests a weak negative relationship between oldpeak (ST segment depression after exercise) and thalach (maximum heart rate achieved during exercise). This implies that as oldpeak increases, thalach tends to decrease, and vice versa.

2.2. Cross Validation

Before proceeding with model training, the dataset is divided into two distinct sets: the training data and the test data. The training data, constituting 75% of the dataset, is utilized for model training, allowing the model to learn from the data's patterns and relationships. The remaining 25% of the data is reserved for the test data, which serves as a benchmark to assess the model's ability to make accurate predictions on new, unseen data.

```
> RNGkind(sample.kind = "Rounding")

Warning message:

In RNGkind(sample.kind = "Rounding") : non-uniform 'Rounding' sampler used

> set.seed(231)

> index <- sample(x = nrow(heart), size = nrow(heart)*0.75)

> heart_train <- heart[index,]
```

```
> heart_test <- heart[-index,]

> prop.table(table(heart_train$target))

        0         1

0.5091146 0.4908854
```

With these relatively balanced proportions, it can be inferred that the class distribution in the model remains fairly balanced. This implies that both class 0 and class 1 maintain a comparable representation within the dataset.

2.3. Model Building

This project is a critical phase where we delve into constructing predictive models to gain valuable insights into heart disease diagnosis. This section is subdivided into three core components: Scaling, Logistic Regression, and K-Nearest Neighbour.

2.4. Scaling

Scaling plays a pivotal role in ensuring the accuracy and effectiveness of the k-Nearest Neighbours (kNN) algorithm in this project. Specifically, z-score standardization is employed to prepare the dataset for kNN classification.

Using the summary() function to obtain summary statistics for each variable in dataset, which includes information such as minimum, 1st quartile, median (2nd quartile), mean, 3rd quartile, and maximum values. This will help understand the range of values in your dataset.

```
> summary(heart)
```

	age	sex	cp	trestbps	chol	fbs
Min. :	29.00	0:312	0:497	Min. : 94.0	Min. :126	0:872
1st Qu.:	48.00	1:713	1:167	1st Qu.:120.0	1st Qu.:211	1:153
Median :	56.00		2:284	Median :130.0	Median :240	
Mean :	54.43		3: 77	Mean :131.6	Mean :246	
3rd Qu.:	61.00			3rd Qu.:140.0	3rd Qu.:275	
Max. :	77.00			Max. :200.0	Max. :564	

```
restecg     thalach          exang       oldpeak         slope    ca          thal
0:497   Min.   : 71.0       0:680    Min.   :0.000   0: 74   0:578       0:  7
1:513   1st Qu.:132.0       1:345    1st Qu.:0.000   1:482   1:226       1: 64
2: 15   Median :152.0                Median :0.800   2:469   2:134       2:544
        Mean   :149.1                Mean   :1.072           3: 69       3:410
        3rd Qu.:166.0                3rd Qu.:1.800           4: 18
        Max.   :202.0                  Max.   :6.200
target
0:499
1:526
```

The analysis shows that the dataset has a wide range of values for various variables, indicating that scaling is necessary to ensure that variables have a similar influence in the modeling process.

Before applying scaling, it's crucial to separate the data into two distinct parts: the predictor variables, which will be used for modelling, and the target variable, which aim to predict. This separation is essential to ensure that the scaling process doesn't impact the target variable, allowing us to maintain the integrity of the predictive relationship during kNN modelling.

```
> heart_train_x <- heart_train %>% select_if(is.numeric)
> heart_test_x <- heart_test %>% select_if(is.numeric)
> heart_train_y <- heart_train[,"target"]
> heart_test_y <- heart_test[,"target"]
> heart_train_xs <- scale(heart_train_x)
> heart_test_xs <- scale(heart_test_x, center = attr(heart_train_xs, "scaled:center"), scale = attr(heart_train_xs, "scaled:scale"))
```

2.5. Models / Algorithms

2.5.1. Logistic Regression

Logistic Regression using backward feature selection is a model-building process that starts with all available predictor variables and iteratively removes the least significant variables until a subset with the most influential variables remains. This technique helps simplify the model, potentially improving its performance and interpretability while retaining essential predictive factors.

i. Build Model

```
> heart_pred_lr <- glm(target ~ ., data = heart_train, family = "binomial")

> heart_model_step <- step(object = heart_pred_lr, direction = "backward",trace = F)
```

ii. Summary Model

```
> summary(heart_model_step)

Call:

glm(formula = target ~ age + sex + cp + trestbps + chol + thalach +

    exang + oldpeak + slope + ca + thal, family = "binomial",

    data = heart_train)

Coefficients:

             Estimate Std. Error z value Pr(>|z|)

(Intercept)  0.133489   2.486789   0.054 0.957191

age          0.029109   0.015923   1.828 0.067534 .

sex1        -2.220298   0.363135  -6.114 9.70e-10 ***

cp1          1.122160   0.370357   3.030 0.002446 **

cp2          2.027204   0.324136   6.254 4.00e-10 ***

cp3          2.443180   0.446890   5.467 4.58e-08 ***

trestbps    -0.026347   0.007310  -3.604 0.000313 ***

chol        -0.006279   0.002576  -2.437 0.014799 *

thalach      0.027332   0.007788   3.509 0.000449 ***
```

```
exang1      -0.830875   0.282849  -2.938 0.003308 **
oldpeak     -0.426937   0.144101  -2.963 0.003049 **
slope1      -0.978950   0.540180  -1.812 0.069945 .
slope2       0.346692   0.584982   0.593 0.553412
ca1         -2.252339   0.318190  -7.079 1.46e-12 ***
ca2         -3.630050   0.514034  -7.062 1.64e-12 ***
ca3         -1.967510   0.573945  -3.428 0.000608 ***
ca4          1.574593   0.978595   1.609 0.107609
thal1        3.026894   1.911558   1.583 0.113314
thal2        2.141184   1.862130   1.150 0.250203
thal3        1.131188   1.866392   0.606 0.544460
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 1064.42  on 767  degrees of freedom
Residual deviance:  465.72  on 748  degrees of freedom
AIC: 505.72
Number of Fisher Scoring iterations: 6
```

iii. Interpretation Model

The model results indicate the relationship between the target variable (presence or absence of heart disease) and several predictor variables. The coefficients associated with each predictor variable represent their impact on the log-odds of the target variable.

Notable findings from the model include:

- ***age*** has a positive coefficient, suggesting that as a person's age increases, the likelihood of having heart disease also increases.
- ***sex*** with a value of 1 (indicating male) has a significant negative coefficient, implying that being male is associated with a lower likelihood of heart disease.
- ***cp*** (chest pain type) are significant predictors, with positive coefficients, indicating a positive correlation with heart disease.
- ***trestbps*** (resting blood pressure) and `chol` (serum cholesterol) have negative coefficients, suggesting that higher values of these variables are associated with a lower likelihood of heart disease.
- ***exang1*** (presence of exercise-induced angina) has a negative coefficient, indicating a reduced likelihood of heart disease when angina is present during exercise.
- ***ca*** variable (number of major vessels coloured by fluoroscopy) are significant predictors. Higher values of ca are associated with a lower likelihood of heart disease.

iv. Prediction

```
> heart_test$pred_lr <- predict(heart_model_step, heart_test, type = "response")

> heart_test$pred_label_lr <- ifelse(heart_test$pred_lr >= 0.5, yes = 1, no = 0)

> heart_test %>% select(target, pred_lr, pred_label_lr) %>% rmarkdown::paged_table()
```

	target	pred_lr	pred_label_lr
4	0	0.1348006506	0
5	0	0.0942410995	0
7	0	0.0838275258	0
8	0	0.0029430923	0
13	1	0.9948034146	1
19	1	0.9886080332	1
20	1	0.9737352985	1
23	1	0.1509253263	0

26	0	0.2780940867	0
27	1	0.9377136080	1
29	0	0.2947696141	0
30	0	0.0481737867	0
31	0	0.4769137698	0
37	1	0.3620628370	0
43	0	0.9712212632	1
47	1	0.7413706476	1
50	0	0.0167133433	0
57	1	0.7992820883	1
58	1	0.8124024714	1
61	1	0.9601660183	1
64	1	0.9220497047	1
77	1	0.5275600593	1
83	0	0.3822461696	0
85	1	0.9971579669	1
87	1	0.9737352985	1
89	0	0.1104813713	0
91	1	0.9980979113	1
92	1	0.9866379426	1
93	0	0.0028130324	0
102	1	0.6737050744	1
105	1	0.9921096318	1

109	0	0.5243669302	1
116	0	0.2780940867	0
120	1	0.8965638340	1
126	1	0.9964653306	1
127	1	0.4737891438	0
131	1	0.9964653306	1
134	1	0.9760927126	1
135	1	0.8268110136	1
136	0	0.0244142173	0
153	0	0.0450457111	0
163	0	0.2828536600	0
164	0	0.2312628549	0
168	1	0.9058702511	1
169	1	0.4112509555	0
170	1	0.9719165347	1
172	0	0.0381859776	0
182	1	0.8168239305	1
185	1	0.8978164362	1
186	0	0.0043099629	0
191	1	0.9818989157	1
195	0	0.0041831577	0
196	1	0.9220497047	1
197	0	0.2581013415	0

209	1	0.9939565169	1
212	0	0.0072817521	0
221	0	0.8420484863	1
226	1	0.8034593345	1
229	1	0.9679022461	1
231	0	0.0450457111	0
232	1	0.9540588680	1
238	0	0.7295454491	1
241	1	0.7844535775	1
242	0	0.6142708223	1
246	1	0.9586001649	1
249	1	0.9732827718	1
254	0	0.0043099629	0
258	1	0.7463930991	1
259	0	0.3831541810	0
263	1	0.9066792555	1
267	0	0.0002970236	0
272	1	0.8467604190	1
274	0	0.0247060710	0
276	0	0.0381859776	0
280	1	0.9632693533	1
291	1	0.9939373130	1
303	1	0.9768889306	1

306	0	0.0005575060	0
308	1	0.9482118073	1
314	1	0.7659301046	1
316	1	0.3843490991	0
322	1	0.9901239474	1
325	1	0.8124024714	1
327	0	0.1580155401	0
330	1	0.9216275468	1
336	0	0.0625883721	0
342	1	0.9769654158	1
343	1	0.9886979839	1
346	0	0.0932831970	0
350	0	0.2549770870	0
351	1	0.4737891438	0
353	0	0.0032283080	0
356	1	0.6892537671	1
364	1	0.8534858902	1
367	0	0.1940346732	0
374	0	0.6331238547	1
376	1	0.5867905557	1
377	1	0.8600831824	1
378	1	0.0349464594	0
381	0	0.0033412467	0

390	1	0.9548539066	1
393	1	0.9535362310	1
398	0	0.0360589178	0
404	1	0.9967835603	1
406	0	0.2947696141	0
407	1	0.9737352985	1
411	1	0.9632693533	1
412	0	0.0216458492	0
422	1	0.7764848577	1
424	0	0.5914721633	1
436	1	0.9058702511	1
439	1	0.9752796519	1
444	1	0.9590355793	1
447	1	0.8534858902	1
448	0	0.0029430923	0
449	1	0.9942576146	1
450	0	0.4110702346	0
452	1	0.9543437176	1
456	1	0.2947958869	0
462	1	0.5275600593	1
466	1	0.9939565169	1
470	0	0.0041831577	0
475	0	0.8009941281	1

476	1	0.6737050744	1
477	0	0.0043099629	0
479	1	0.9444739741	1
482	0	0.0207518602	0
487	0	0.7160222271	1
491	1	0.7500828256	1
492	1	0.9590355793	1
497	0	0.1412411694	0
499	1	0.6557647058	1
501	1	0.9010932265	1
508	1	0.9870591133	1
516	0	0.1364185089	0
518	1	0.9632693533	1
523	1	0.9645521156	1
525	0	0.1940346732	0
526	1	0.9930330167	1
531	0	0.0181863672	0
535	1	0.9980979113	1
539	0	0.4011243026	0
541	0	0.0005907641	0
542	1	0.9696653207	1
543	1	0.6718924993	1
544	0	0.5514269750	1

547	0	0.0127819518	0
551	0	0.1178235453	0
554	1	0.9311450845	1
567	1	0.9807260741	1
570	1	0.9299799040	1
572	0	0.0005236161	0
581	1	0.9018543162	1
588	0	0.0047515904	0
594	0	0.1176090044	0
598	1	0.9939565169	1
600	1	0.8728915406	1
602	0	0.0002783160	0
608	0	0.0717422568	0
618	1	0.9844953123	1
619	1	0.9065169157	1
622	0	0.0098990436	0
626	0	0.1178235453	0
627	0	0.1145046370	0
629	1	0.8337441792	1
633	1	0.9664406759	1
634	0	0.0064911660	0
647	0	0.9728686786	1
648	1	0.0349464594	0

650	1	0.9664406759	1
651	0	0.0005575060	0
652	1	0.9719044871	1
653	1	0.9635181135	1
656	1	0.9258574778	1
657	0	0.8420484863	1
664	1	0.8454966631	1
667	1	0.9467420447	1
670	0	0.0932831970	0
676	0	0.2581013415	0
678	0	0.0188366565	0
689	0	0.0012093558	0
690	1	0.9955091482	1
691	1	0.6524051123	1
698	0	0.6142708223	1
702	0	0.0362127476	0
710	0	0.5885980319	1
713	1	0.9719165347	1
723	1	0.9645521156	1
724	1	0.9650618099	1
727	0	0.2312628549	0
733	1	0.5316495290	1
738	0	0.0013198292	0

743	0	0.0143253570	0
745	1	0.9787878577	1
752	1	0.9791692558	1
753	1	0.8124024714	1
754	1	0.6605698205	1
760	0	0.9712212632	1
762	1	0.9673476310	1
763	1	0.8563951733	1
764	1	0.9934693626	1
769	1	0.9018543162	1
776	1	0.7546938980	1
780	1	0.9948034146	1
787	0	0.0157840306	0
788	0	0.0006978933	0
791	0	0.0036976643	0
800	1	0.4574525673	0
805	1	0.8383908975	1
809	1	0.5316495290	1
810	1	0.9864572975	1
814	0	0.0916222977	0
817	1	0.8606685265	1
828	0	0.2007225937	0
829	1	0.9377136080	1

838	1	0.7965809089	1
845	0	0.0054195470	0
846	0	0.0154921789	0
848	0	0.0064911660	0
852	1	0.8557615642	1
857	1	0.9650618099	1
859	1	0.9650618099	1
860	0	0.1860429464	0
872	1	0.9444739741	1
880	1	0.8526330265	1
887	0	0.0031827617	0
903	0	0.1104813713	0
906	0	0.0149568713	0
907	1	0.7991652043	1
919	0	0.5514269750	1
924	1	0.6557647058	1
928	1	0.9299799040	1
930	0	0.0011300512	0
933	1	0.8174265469	1
940	1	0.9543437176	1
950	1	0.5084024326	1
952	0	0.2549770870	0
955	1	0.8165919112	1

970	1	0.9205806984	1
972	1	0.7965809089	1
974	1	0.7803830528	1
975	1	0.6942670438	1
978	1	0.9679022461	1
984	1	0.0045226753	0
988	0	0.4746161950	0
991	1	0.9540588680	1
995	0	0.0129912862	0
996	1	0.8467604190	1
1000	0	0.0157840306	0
1008	1	0.7992820883	1
1012	1	0.8978164362	1
1013	0	0.8009941281	1
1014	0	0.0838275258	0
1015	1	0.9955029094	1
1018	0	0.0002970236	0
1022	0	0.0039652194	0
1025	0	0.0113591703	0

2.5.2. K-Nearest Neighbor

i. find optimum k

```
> sqrt(nrow(heart_train_xs))
```

[1] 27.71281

The choice of *k* can significantly impact the model's performance. To find the ideal *k*, a common approach is to calculate sqrt(nrow(data)) which helps provide a starting point for *k* selection. In this case, this calculation yields approximately 27.71281. Since the dataset contains binary target classes (0 and 1), it is advisable to opt for an odd value of *k* to avoid potential ties in the voting process. Therefore, setting *k* to 27, as suggested by the calculation.

ii. Build Model

```
> heart_pred_knn <- knn(train =  heart_train_xs, test = heart_test_xs, cl = heart_train_y, k = 27)
```

3. Results

3.1. Evaluation

3.1.1. Logistic Regression

```
> confusionMatrix(data = as.factor(heart_test$pred_label_lr), reference = heart_test$target, positive = "1")
```

Confusion Matrix and Statistics

	Reference	
Prediction	0	1
0	91	11
1	17	138

Accuracy : 0.8911

95% CI : (0.8464, 0.9264)

No Information Rate : 0.5798

P-Value [Acc > NIR] : <2e-16

Kappa : 0.7747

Mcnemar's Test P-Value : 0.3447

Sensitivity : 0.9262

Specificity : 0.8426

Pos Pred Value : 0.8903

Neg Pred Value : 0.8922

Prevalence : 0.5798

Detection Rate : 0.5370

Detection Prevalence : 0.6031

Balanced Accuracy : 0.8844

'Positive' Class : 1

3.1.2. K-Nearest Neighbour

> confusionMatrix(data = heart_pred_knn, reference = heart_test_y, positive = '1')

Confusion Matrix and Statistics

Reference

Prediction	0	1
0	69	23
1	39	126

Accuracy : 0.7588

95% CI : (0.7017, 0.8098)

No Information Rate : 0.5798

P-Value [Acc > NIR] : 1.467e-09

Kappa : 0.4946

Mcnemar's Test P-Value : 0.05678

Sensitivity : 0.8456

Specificity : 0.6389

Pos Pred Value : 0.7636

```
     Neg Pred Value : 0.7500
          Prevalence : 0.5798
      Detection Rate : 0.4903
Detection Prevalence : 0.6420
 Balanced Accuracy : 0.7423
       'Positive' Class : 1
```

3.2. Comparison

Metric	Logistic Regression	K-Nearest Neighbour
Accuracy	0.8911	0.7588
Sensitivity	0.9262	0.8456
Specificity	0.8903	0.6389

The comparison of the confusion matrix that represents the classified TP values, FP values, FN values and TN values shows that the False negative of 11 and false positive of 17 in case of Logistic Regression approach depicts a better result than that of K-NN approach with a False negative of 23 and False positive of 39.

4. Conclusion

Following the process, both model performances exhibited notable improvements. A thorough examination of the comparison table reveals that the Logistic Regression model achieved higher scores across crucial metrics, including Recall, Accuracy, Specificity, and Precision, in contrast to the K-Nearest Neighbor (K-NN) model. The pivotal metric to consider depends on the specific objective at hand, and in this context, place significant emphasis on the Recall metric. This is particularly important when we aim to minimize instances where the model incorrectly predicts patients as Target 1 when they are, in fact, healthy, or vice versa. Consequently, the Logistic Regression models emerge as the preferred choice, as they exhibit superior predictive accuracy in identifying individuals with health issues. The Recall value of 92.62% achieved by the Logistic Regression models surpasses the performance of the K-NN models.