Predicting Heart Failure: A Machine Learning Comparative Investigation.

By Justin Grima (14248599)

I. Abstract

Heart Failure (HF) has been a prevalent global pandemic for years, in 2014 “approximately 26 million people worldwide were living with heart failure”[1] (Ponikowski et al., 2014) with this number only increasing in our present day. To help improve these statistics, predictive models for this disease can provide a key approach to decreasing its impact. The **purpose** of this report aims to analyse the “DataClean-fullage"[2] (Coronary Artery Disease - Analysis, n.d.) dataset and **i)** use supervised machine learning algorithms to create predictive models for heart failure, **ii)** create unsupervised machine learning algorithms to analyse and discover interesting natural cluster patterns in the response variable; heart failure and **iii)** use association rule mining to find a list of established association rules for heart failure and its possible association with other medical conditions within the dataset. The **methodological approaches** use the program RStudio to conduct data cleaning, manipulation, supervised (Naïve Bayes and Logistic Regression), unsupervised (Agglomerative Hierarchical Clustering (AHC) and DBSCAN), and descriptive (Association Rule Mining) algorithms, summarizations, and visualisations; using various packages such as ‘cluster’, ‘naivebayes’, ‘MASS’, ‘ROCR’ and ‘arules’ to conduct on the data set ‘DataClean-fullage’ to investigate the topic of heart failure.

The study's **findings** demonstrate the superiority of logistic regression combined with stepwise regression; used to create a subset of independent variables (from the main dataset) significantly influencing the dependent variable: heart failure, to predict the classification of future observations (patients). AHC’s Ward’s method provided insights into the natural clusters pertaining to heart failure, concluding that a distinct cluster of two groups produced the best results, supporting the outcomes of the response variable (heart failure and no heart failure). Using association rule mining unfortunately established no association rules for heart failure and other medical conditions. Having performed the analysis, we can **conclude** that with the subset of variables from the ‘DataClean-fullage’ dataset, through unsupervised machine learning clustering algorithm analysis that the most distinct natural clustering size is 2 using AHC Ward method, which had the best performance, and aligns what we already know about the Heart Failure response variable from the original dataset. In our supervised machine learning algorithms, we found that training and testing a logistic regression model resulted in the best performance with the highest accuracy when predicting observation with the test data and is recommended to use to predict future outcomes of heart failure. Finally using the association rule mining, we were not able to find any rules, but with that said, as more data is collected and possible variables added, it may be possible in the future.

II. Introduction

Heart failure is a long-term “condition where your heart muscle doesn't pump blood as well as it should”[3] (Healthdirect Australia, 2020) possibly stemming from “several health conditions, your lifestyle, and your age and family history”[4] (CDC, 2019) that may contribute to increasing your risk. Regardless of the underlying causes, heart failure provides a global burden as it affects “at least 26 million people worldwide and is increasing in prevalence”[5] (Savarese & Lund, 2017) placing great stress on not only the patients but caregivers and the healthcare system. In the past, it has been estimated that heart failure has resulted in a “health expenditure of around $31 billion”[6] (Mozaffarian et al., 2016) with the estimated cost expected to “increase by 127 % between 2012 and 2030.” [6] (Mozaffarian et al., 2016). The projected increase by such an alarming rate can be contributed to the possible lack of awareness of the disease or not taking the issue seriously which has and will continue to result in premature deaths globally.

Society needs to improve in providing better awareness, incentivising the promotion of self-care through a healthy lifestyle, and continuing to conduct research to combat this global issue that claims so many lives. In the past, researchers have embarked on studies to collect and analyse data with the objective of predicting and treating heart failure. These datasets can be found through a vast number of resources online; from UCI Machine Learning Repository, Kaggle, to global electronic health records of patients who have had heart failure. From these records and datasets is possible to conduct research into exploratory character analysis of the disease, and the global citizens it affects, as well as using supervised machine learning classification algorithms to predict future outcomes. This report aims to do just that, conduct exploratory data analysis, supervised, unsupervised and descriptive machine learning algorithms to gain insight into characteristics, natural clustering of the response variable: heart failure, prediction of heart failure and its associative rules with other variables corresponding to other medical conditions.

III. Data

As previously mentioned in the abstract above, to conduct the analysis we will be using the ‘DataClean-fullage.csv’ dataset retrieved from Kaggle; “a subsidiary of Google LLC, is an online community of data scientists and machine learning practitioners”[7] (Wikipedia Contributors, 2019) which is a subset of the original dataset ‘HDHI Admissions Data.csv’ which is an observational study where the “data was collected from patients admitted over a period of two years (1 April 2017

to 31 March 2019) at Hero DMC Heart Institute, Unit of Dayanand Medical College and Hospital, Ludhiana, Punjab, India. This is a tertiary care medical college and hospital. During the study period, the cardiology unit had 14,845 admissions corresponding to 12,238 patients. 1921 patients who had multiple admissions."[8] ("Hospital Admissions Data," n.d.). The 'DataClean-fullage.csv' dataset has 53 variables; 7 character variable, 11 numerical (continuous) variable, 35 numeric binary categorical variables, and 6611 observations. Missing values and duplicates were checked for with 0 found. With that said the 'DataClean-fullage.csv' data set has had pre-processing done where the original dataset; 'HDHI Admissions Data.csv', contained the variables **i)** Cardiogenic Shock: Patients in shock due to cardiac reasons, **ii)** Shock: Systolic blood pressure < 90 mmHg, and when the cause for shock was any reason other than cardiac, **iii)** Month of the year, **iv)** Duration of stay, **v)** Admission number, **vi)** Date of admission, **vii)** Date of Discharge and **viii)** Housing location. There is no reason provided as to why they were removed, it can be assumed that they were deemed not relevant in the researchers' research study as this holds true for this study.

The 'HDHI Admissions Data.csv' dataset had 15757 observations with the 'DataClean-fullage.csv' having 6611 observations, as mentioned before. Again, there was no reasoning provided as to why so many observations had been removed. Upon investigation, there were no duplicates (specifically checking the serial number and Admission number identification variables), therefore the assumption for removing duplicates as a reason for removing observation said observations does not seem like a feasible explanation (some assumptions for supervised machine learning algorithms require observation to be independent of one another, therefore removing duplicates could resolve this assumption violation, but is not a recommended form of practice). An alternative reasoning would be to have a balanced response variable outcome. In the 'HDHI Admissions Data.csv' dataset, the outcome for having and not having heart failure is 28.94% vs 71.05%. Therefore, it is possible to assume observations were removed to make the data more balance since the outcome for having and not having heart failure is 47.78%% vs 52.22% in the 'DataClean-fullage.csv' dataset. (See RMarkdown Part 2 for raw code.)

IV. Methods

The following analysis was conducted in RStudio: Spotted Wakerobin, version 2022.07.2+576. Within the dataset, several variables can be used as a response variable for supervised machine learning algorithms, which one can argue makes this a multipurpose dataset. In this study, we will focus on heart failure as the response variable. This investigative report requires the implementation of at least three of the five listed machine-learning algorithm categories. Before further discussing the methods used, it is important to note that stepwise regression (which will be discussed later) was conducted to reduce the size of the dataset and find a set of independent variables that significantly influence the dependent variable. Therefore, when referring to 'the data subset' it is referring to this process that occurred. The first category involves Linear Discriminant Analysis (LDA), Quadradic Discriminant Analysis (QDA), or Naive Bayes. Both LDA and QDA require the predictor variables to be numerical. Although two of the variables in the subset are numerical, the remaining are binary categorical variables. Therefore, the assumption is not met for QDA and LDA and cannot be used. Naïve Bayes was chosen because the subset is more suited for this algorithm, which will be discussed in detail further in this investigation.

In the second category, we have the option for Logistic Regression classifiers and/or K-Nearest Neighbour (KNN) for classification/regression. In the data set the dependent (response) variable 'HeartFail' is binary. Due to the nature of the dependent variable, our first choice for a supervised machine learning algorithm would be Binary Logistic Regression because it is a direct way of estimating the probability of a binary response variable; we assume an appropriate probability distribution for the binary class labels $Yi = \{0,1\}$. Therefore, based on the characteristics of the response variable, we are more inclined to use logistic regression which was chosen over KNN method. In addition, using logistic regression will allow the use of the stepwise method to reduce the large dataset where through a series of tests (e.g. F-tests, t-tests) it can find a set of independent variables that significantly influence the dependent variable. An alternative to stepwise regression would have been Principal Component Analysis (PCA; the third category option) which is most notably used in data mining for dimensionality reduction and data visualisation. Unfortunately, due to the requirements of PCA; using numerical predictor variables, and the characteristics of the original dataset; having mixed variables (numerical and categorical variables), PCA was not used.

Regarding unsupervised machine learning algorithms; Cluster Analysis (the fourth category), there are three popular cluster analysis techniques: K-means, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and its sister algorithm HDBSCAN; which doesn't need an epsilon (eps) value specification as opposed to DBSCAN, and Hierarchical clustering; more specifically Algometric Hierarchical Clustering (AHC). When using the k-means algorithm, it assumes that all 'n' variables that describe the data are real-valued; a numeric vector that can be seen as a point in an n-dimensional coordinates space. It is not applicable to categorical data because categorical variables are discrete and do not have any natural origin, so computing Euclidean distance (distance from a centroid as a definition for a cluster) for such space is not meaningful when

using categorical variables. Since K-means can only be used in data sets where you can compute the arithmetic mean (mean is a least-squares estimator) which minimizes the squared Euclidean, this puts severe limits on what distances can be used with k-means. Since there are mixed variables in the dataset, the distance that would be used is Gowers distance, which, if you were to supply K-means with a distance object, it would treat it as a data frame, and the outcomes would be incorrect unlike the hclust() function in AHC and the dbscan() function in DBSCAN/HDBSCAN. Therefore k-means cannot be used as a clustering algorithm.

As mentioned, in the case of distance calculation, we would use Gowers to calculate the distance/dissimilarity between rows when the variables are not the same class, for which (as briefly discussed above) can be used in a hclust() in AHC and the dbscan() function in DBSCAN/ HDBSCAN. DBSCAN/ HDBSCAN does its cluster analysis based on the density of the regions where it uses epsilon (ε; the radius of the circle) and minimum points (minPts; minimum number of points inside the circle). For DBSCAN to work it needs numerical variables which, as we've seen, the subset has mixed variables and the fix is to use a Gower distance object instead of the data subset. With that said DBSCAN is a good algorithm for finding outliers in a data set and finding arbitrarily shaped clusters based on the density of data points in different regions but requires there to be a drop in the density of data points to detect boundaries between clusters. This can be a foreseen issue due to the possibility of the dataset not having many drops in density between clusters (i.e. where there is a possibility of clusters overlapping, multiple clusters might get grouped together into one large cluster). These are things to consider when running the DBSCAN and interpreting its results.

In AHC every observation is a cluster on its own and observations are merged with the pair of clusters that are the most similar (least dissimilar) according to a given dissimilarity measure between clusters. When computing the distance/dissimilarity, we use the dist() function within the hclust() function, which will automatically provide the Gowers distance as the method since it recognizes mixed variables within the data subset. AHC is a clustering method that is widely used in many fields such as biology; the essential component of biological data, where the idea that similarity exists on multiple levels, as the similarity is a naturally ordered characteristic and this methodology is not concentrated on when conducting the other clustering analysis that has been mentioned above. Therefore, AHC is the most widely used algorithm for expression data as it specifically addresses similarity on multiple levels, in a very simple manner. The simplicity comes in the form of a visual result when the algorithm completes; a dendrogram. The dendrogram allows us to easily interpret and find clusters and gives us a clear visual indication of its model performance. We can further explore the dendrogram at varying cluster sizes by cutting the dendrogram at specific heights (similarity level) to gain a detailed understanding of the model performance at that cut level and view it on a two-dimensional plot. Overall, the cluster analysis methods we will use are AHC (the preferred method of cluster analysis for the reasons mentioned above) and a brief look at DBSCAN/ HDBSCAN to compare their performances.

Finally, we will use Association Rule Mining (ARM) (category five). The premise of ARM is to analyze data for patterns, or co-occurrences, in a database where it identifies frequent 'if-then' associations, which themselves are the association rules. An association rule has two parts: an '(A)ntecedent' (if/left-hand side; lhs) and a '(C)onsequent' (then/right-hand side; rhs). An antecedent is an item found within the data. A consequent is an item found in combination with the antecedent. Therefore, pertaining to this dataset, the intended outcome of using the association rule mining is to find a list of established rules where the consequent is having heart failure and the antecedents are other possible variables within the dataset that are other diseases. The outcome of this would be to help patients and doctors become aware of other possible medical conditions a patient could or may have that can be tested for and/ or be made aware of if they had heart failure.

V. Results and discussion

Supervised Machine Learning Algorithm: Logistic Regression

As mentioned in the methods, due to the nature of the dependent variable, our first choice for a supervised machine learning algorithm would be Binary Logistic Regression because it is a direct way of estimating the probability of a binary response variable.

```
Call:
glm(formula = HeartFail ~ ., family = "binomial", data = HeartFailure)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.0833  -0.0092  -0.0006   0.0076   3.9303

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)               0.9776688  9.5963252   0.102  0.91885
Age                      -0.1219038  0.1064198  -1.145  0.25200
GenderM                  -0.2429850  1.1449383  -0.212  0.83193
AdmissTypeO              -0.4490071  1.2305609  -0.365  0.71520
ICUDur                    0.1167388  0.0894288   1.305  0.19176
OutcomeDISCHARGE          1.1928452  1.7436761   0.684  0.49391
OutcomeEXPIRY             0.8141568  1.9768098   0.412  0.68045
Smoking                   0.8833030  2.6075001   0.339  0.73479
```

Figure 1a: Summary of logistic regression with the original data set.

```
GroupLeuklow              -0.0805810  3.9120360  -0.021  0.98357
GroupLeuknormal           -0.5029998  1.0818748  -0.465  0.64198
GroupEjectFracb_mildHF     1.6782596  6.0956889   0.275  0.78307
GroupEjectFracc_belowNormal 0.4875447  7.3989932   0.066  0.94746
GroupEjectFracd_normal     2.8145756  3.3080381   0.851  0.39486
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 9151.802  on 6610  degrees of freedom
Residual deviance:   64.839  on 6552  degrees of freedom
AIC: 182.84

Number of Fisher Scoring iterations: 13
```

Figure 1b: Summary of logistic regression with the original data set.

Figures 1a and 1b provide a summary of the logistic regression on the whole dataset, where we can see that there are a lot of predictor variables (as seen in the EDA; 53 variables) where their p-values > 0.05 which indicates no statistical significance of these predictor variables to the model (these essential correlates to theses variables beta zeros (y-intercept): 'Estimate', in the summary being equivalent to 0). There are a few ways to deal with enough insignificant variables in the model where we can fit a better model with fewer parameters. Two main methods are the Principal Component analysis (PCA) and the stepwise selection (regression) method. As mentioned, we cannot use PCA because it requires the predictor variable to be numerical. In the original dataset, there is the presence of a few numerical variables (9 out of 53; ~17%,), but most of the variables are categorical; nominal and ordinal, and some character class. Therefore, there isn't a significant number of numerical variables in the dataset to use PCA. The alternative then is to use the stepwise selection (regression) method which consists of iteratively adding and removing predictors, in the predictive model, to find a subset of variables in the data set resulting in the best performing model, which is a model that lowers prediction error and finds significant variables.

```
Call:
glm(formula = HeartFail ~ Age + Alcohol + Urea + HFRedEjeFrac +
    HFNormEjeFrac + ValvHeaDis + ContHeaDis, family = "binomial",
    data = HeartFailure)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.9760  -0.0228  -0.0062   0.0281   4.0466

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)   -5.221936   1.779489  -2.935  0.00334 **
Age           -0.040634   0.027187  -1.495  0.13502
Alcohol       -2.852395   1.232421  -2.314  0.02064 *
Urea          -0.010457   0.005793  -1.805  0.07106 .
HFRedEjeFrac  15.952280   1.451589  10.990  < 2e-16 ***
HFNormEjeFrac 16.767213   1.619701  10.352  < 2e-16 ***
ValvHeaDis     4.062709   1.231458   3.299  0.00097 ***
ContHeaDis    -5.388179   1.361022  -3.959 7.53e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 9151.802  on 6610  degrees of freedom
Residual deviance:   84.153  on 6603  degrees of freedom
AIC: 100.15

Number of Fisher Scoring iterations: 11
```

Figure 3: Stepwise selection (regression); backward direction summary.

Figure 3 depicts the relevant variables found because of the stepwise selection (regression): **I)** Age (Numerical, Continuous): Age of the patients. **II)** Alcohol (Binary, Categorical): Consumption of alcohol. **III)** Urea (Numerical, Continuous): Formed in the liver via the urea cycle from ammonia and is the product of protein metabolism. Healthy kidneys filter urea and remove other waste products from your blood. **IV)** Heart Failure with Reduced Ejection Fraction (Binary, Categorical): Heart Failure with 39% or less ejection fraction of the left ventricle. **V)** Heart Failure with Normal Ejection Fraction (Binary, Categorical): Heart Failure with 40% or above ejection fraction of the left ventricle. **VI)** Valvular Heart Disease (Binary, Categorical): When any valve in the heart has damage or is diseased and **VII)** Congenital Heart Disease (Binary, Categorical): An abnormality in the heart that develops before birth. The 'null deviance' (a goodness-of-fit metric) in the output is from fitting a model with no predictors; this is the worst possible model, that says the response does not depend on any of the predictors (Null deviance = 9151.802). The 'residual deviance' is significantly different to the null deviance as it shows how well the response is predicted by the model when the predictors are included (Residual deviance = 84.153). The lower the value, the better. Observing the AIC scores where lower AIC values indicate a better-fit model, the reduced model has an AIC=100.15 compared to the full model's (Figure 1b) AIC=182.84. Comparing the residual deviance using the subset we have D=84.153, versus the full dataset D=64.839. This shows that there wasn't much cost to reducing the number of predictors since the Residual Deviance was higher in the full model, but overall, the AIC was significantly reduced in the stepwise selection (regression), backward direction. Since the AIC score for a model is equal to the residual deviance plus two times the number of parameters (K) when comparing models, the one with the smaller AIC score is better supported by the data. Therefore, the model using the stepwise selection, where the outcome involves the variables mentioned before, is the better choice.

From the model's summary, we can look at the statistical significance of the predictor variables to the response variables. We see that 'Alcohol', 'HFRedEjeFrac', 'HFNormEjeFrac', 'ValvHeaDis', and 'ContHeaDis' have a p-value < 0.05 which correlates to statistical significance. Both 'Urea' (0.07106) and Age (0.13502) have values where p-value > 0.05 which means they are not statistically significant but were still found to be important contributions to the model. To validate the selection of the two variables, we can investigate their correlation to heart failure. Research has shown that as your age increases there is a high chance of heart failure because the is a deterioration

in cardiac structure and function which leads to increased susceptibility to heart failure. Urea is also found to be significant because urea nitrogen is a waste product that your kidneys remove from your blood. If there is a constant presence of high blood urea nitrogen levels, this correlates to an increased risk of a cardiovascular episode for patients with acute heart failure (which means that the heart is still beating, but it is not able to deliver the required oxygen to meet the body's needs). Therefore, the justification for using these two variables is sound and overall, the results from the stepwise selection (regression) method provide concrete results to consider using these variables as a subject to the main dataset to use for the remainder of this study (See RMarkdown Part 3 for raw code).

```
Call:
glm(formula = HeartFail ~ ., family = "binomial", data = HF_LR_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.8418  -0.0281  -0.0192   0.0373   3.9524

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -7.031407   2.233979  -3.147 0.001647 **
Age           -0.009042   0.032843  -0.275 0.783080
Alcohol       -0.109645   2.678300  -0.041 0.967345
Urea          -0.006162   0.008390  -0.734 0.462663
HFRedEjeFrac  15.066727   1.425182  10.572  < 2e-16 ***
HFNormEjeFrac 15.408853   1.476660  10.435  < 2e-16 ***
ValvHeaDis     3.336402   1.418871   2.351 0.018700 *
ContHeaDis    -4.948989   1.385711  -3.571 0.000355 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7321.666  on 5288  degrees of freedom
Residual deviance:   67.508  on 5281  degrees of freedom
AIC: 83.508

Number of Fisher Scoring iterations: 10
```

Figure 4: Summary of logistic regression with subset data set.

For supervised machine learning algorithms we split the subset data into training (80% of the dataset) and test data (the remaining 20%). With the training data, we can train the logistic regression model to learn how to classify the outcome of heart failure for future observations. We then use the test data to test the model and calculate its accuracy in correctly predicting the correct classification. Figure 4 shows the summary of the logistic regression with the training data of variables chosen from the stepwise regression. We can see residual deviance = 67.508 and an AIC score of 83.508. The 'Age', 'Alcohol' and 'Urea' variables have p-values > 0.05 which would mean no statistically significant but really correlates to the ‘Estimate’ (beta 0; y-intercept) in the summary essentially being 0 for these variables. For the remaining variables; ‘HFRedEjeFrac’, ‘HFRedNormFrac’, ‘ValvHeaDis’ and ‘ContHeaDAis’ we can see the p-value < 0.05 which correlates to these predictor variables being statistically significant. Overall, compared to figure 1a and 1b, the logistic regression with the subset variables had a better performance (AIC = 83.508) vs using the whole dataset (AIC = 182.84) which reconfirms the validation of using the stepwise regression.

```
       Age      Alcohol         Urea HFRedEjeFrac HFNormEjeFrac    ValvHeaDis
  1.116592     1.013905     1.195644     2.366195     1.719778     1.593078
ContHeaDis
  1.416402
```

Figure 5: VIF scores for logistic regression assumptions.

```
Confusion Matrix and Statistics


HF_LG_pred   0   1
         0 689   1
         1   1 631

               Accuracy : 0.9985
                 95% CI : (0.9945, 0.9998)
    No Information Rate : 0.5219
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.997

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9986
            Specificity : 0.9984
         Pos Pred Value : 0.9986
         Neg Pred Value : 0.9984
             Prevalence : 0.5219
         Detection Rate : 0.5212
   Detection Prevalence : 0.5219
      Balanced Accuracy : 0.9985

       'Positive' Class : 0
```

Figure 6: Confusion matrix for logistic regression.

Figure 5 depicts the variance inflation factor (VIF) scores; a measure of how easily a variable is predicted from a linear regression using the other predictors. A general guideline is that a VIF score larger than 10 indicates that the model has problems estimating the coefficient and is considered an issue that would need to be further investigated. The VIF values in figure 5 are between 1.014 and 2.637 which is a low value that correlates to no multicollinearity, which satisfied one of the assumptions of logistic regression. A second assumption for logistic regression is that the dependent variable; heart failure, is binary which we know to be true as we previously discussed. The

final assumption is that observations are independent of one another. The metadata mentioned that 1921 patients had multiple admissions, but these observations were removed from the original dataset by the researcher from the source from which this dataset was retrieved. This assumption was double-checked by identifying possible duplicates; specifically, the serial numbers recorded in the original dataset, where no duplicates/ matched data was found. There is also no indication of any time series variables that would suggest multiple observations from the same patients occurred over time as well. Therefore, the assumptions for logistic regression are met.

Figure 6 shows the confusion table which explains the outcome of using the trained logistic regression model to classify heart failure using the test data taken from 20 % of the data subset as explained earlier. We can see the accuracy of the model is 99.85%, where the sensitivity value (proportion of positive classifications out of the number of samples that were actually positive) is 99.86%; 689 true negatives (didn't have heart failure) and 1 false positive (predicted as having heart failure, but actually didn't), and the specificity (proportion of negative classifications out of the number of samples which were actually negative) of 98.84%; 631 true positives (had heart failure) and 1 false negative (predicted to not have heart failure, but actually did). Overall, we can conclude logistic regression is a high-performance classification model based on the results shown in figure 6.

At this time, it is worth mentioning that there is an alternative to test accuracy aside from using the confusion table shown in figure 6 and that is the Area Under the Curve (AUC) Accuracy. With the use of simple accuracy metrics, problems can give an inflated sense of confidence in model predictions that is detrimental to some objectives. Alternatively, AUC calibrates the trade-off between sensitivity and specificity at the best-chosen threshold. Further, 'accuracy' measures how well a single model is doing, whereas AUC compares two models as well as evaluates the same model's performance across different thresholds. Both are helpful for comparing and assessing how well a model is doing. When determining which accuracy to use, 'accuracy' (using the confusion table) is a sufficient metric for balanced data, but AUC is well-suited to measure the model's performance on an imbalanced set. When conducting our EDA of the original data set, we found that the response variable outcomes were 52.22 % of patients not having heart failure and 47.78% of patients having heart failure. From this, we can see that the outcome for the response variable are balanced and therefore we will choose the original 'accuracy' (using the confusion table) as the measure of performance (See RMarkdown Part 4 for raw code).

Supervised Machine Learning Algorithm: Naïve Bayes

Another supervised machine learning algorithm that can be used is the Naïve Bayes classifier to estimate the probabilities from the data. Naïve Bays assumes that the dependent variable, Y, and the predictor(s), X, are categorical. With that said we can use numerical predictor variables, but the model no longer estimates the probabilities P(Xi|Y) in a frequentist way, because such predictor(s) can take an infinite number of values. The workaround used by the naive Bayes classifier is to model these probabilities using some form of the probability density function. The most common approach uses the Normal (Gaussian) distribution, the default setting in the Naive Bayes function. The main assumption for Naive Bayes is that the attributes are conditionally independent given the class. This assumption is called class conditional independence. It is made to simplify the computation involved and, in this sense, is considered 'naïve' because practical applications' databases usually contain correlated predictors. Even when the conditional independence assumption is not entirely met, the class probability estimations usually differ enough that the independence assumption error does not change their order. Therefore, the naive Bayesian classifier performs well even in problems where the conditional independence assumption is not entirely true.

	p-value	Outcome
V1V3	1.617642e-01	Independant
V2V5	1.167017e-01	Independent
V2V6	1.167017e-01	Not independent
V2V7	6.309955e-01	Not independent
V2V8	6.486391e-01	Not independent
V5V6	1.103888e-139	Independent
V5V7	6.046651e-02	Not independent
V5V8	5.994775e-01	Not independent
V6V7	9.346069e-01	Not independent
V6V8	5.587721e-02	Not independent
V7V8	1.144790e-02	Independent

Figure 7: Variable independence results.

```
Confusion Matrix and Statistics


PredClassNB   0   1
         0 689   8
         1   1 624

               Accuracy : 0.9932
                 95% CI : (0.9871, 0.9969)
    No Information Rate : 0.5219
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9864

 Mcnemar's Test P-Value : 0.0455

            Sensitivity : 0.9986
            Specificity : 0.9873
         Pos Pred Value : 0.9885
         Neg Pred Value : 0.9984
             Prevalence : 0.5219
         Detection Rate : 0.5212
   Detection Prevalence : 0.5272
      Balanced Accuracy : 0.9929

       'Positive' Class : 0
```

Figure 8: Naïve Bayes (without kernel) prediction summary.

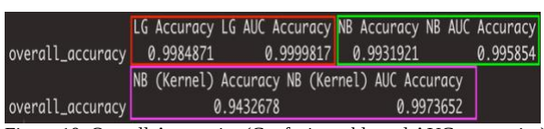
```
Confusion Matrix and Statistics


PredClassNBK   0   1
          0 689  74
          1   1 558

               Accuracy : 0.9433
                 95% CI : (0.9294, 0.9551)
    No Information Rate : 0.5219
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8858

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9986
            Specificity : 0.8829
         Pos Pred Value : 0.9030
         Neg Pred Value : 0.9982
             Prevalence : 0.5219
         Detection Rate : 0.5212
   Detection Prevalence : 0.5772
      Balanced Accuracy : 0.9407

       'Positive' Class : 0
```

Figure 9: Naïve Bayes (with kernel) prediction summary.

From figure 7 variable independence test results, we can conclude that the variable independence assumption is not entirely met. With that said, the hypothesis test (in the Chi-Square test) is suggesting dependence because the null hypothesis is rejected at the 0.05 level. But there is still a possibility that the variables are still independent, and the test could be significant if more data was collected. Regarding independence tests, there is also the fact that there is no obvious hypothesis testing for the independence of the categorical variables with the numerical variables. The purpose of the independence check between the numerical variables and the Chi-Square hypothesis test between categorical variables is to support the statement that the independence assumption is often violated but the fact remains that "naïve Bayes nonetheless often delivers competitive classification accuracy. Coupled with its computational efficiency and many other desirable features, this leads to naïve Bayes being widely applied in practice."[9] (I Webb, 2017) As we in Figures 8&9, although the independence assumption is not entirely met, the model still performs well (See RMarkdown Part 5 for raw code).

As mentioned before, the subset dataset is split 80:20 into training and test data for which the training data is used to train the Naïve Bayes model and the test data is used to test the model's ability to classify the observations into two outcomes; having heart failure and not having heart failure. With Naïve Bayes there is the option to use the kernel function in the model. Kernel Density approximation 'usekernal=T' correlates to the model not being dependent on the normality assumption which can alter the performance/ accuracy of the prediction test results in either a positive or negative way depending on the dataset. Therefore, Naïve Bayes is trained and tested with (figure 8), and without (figure 9) the use of the kernel function. Figure 8 shows the prediction results (using the test data) from the trained Naïve Bayes model without the use of the kernel function. We can see in the summary that the accuracy (based on the confusion table at the top) of the model is 99.32%, where the sensitivity value (proportion of positive classifications out of the number of samples that were actually positive) is 99.86%; 698 true negatives (didn't have heart failure) and 1 false positive (predicted as having heart failure, but actually didn't), and the specificity (proportion of negative classifications out of the number of samples which were actually negative) of 98.73%; 624 true positives (had heart failure) and 8 false negatives (predicted to not have heart failure, but actually did). (See RMarkdown Part 6 for raw code).

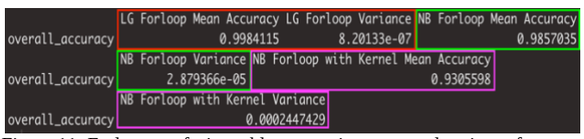
Figure 9 shows the prediction results (using the test data) from the trained Naïve Bayes model with the use of the kernel function. We can see in the summary that can see the accuracy (based on the confusion table at the top) of the model is 94.33%. There is a sensitivity of 99.86%: 689 true negatives and one false positive, and a specificity of 88.29%: 558 true positives and 74 false negatives. We can conclude, in this scenario, that the use of kernel altered the performance/ accuracy negatively. (See RMarkdown Part 7 for raw code).



	LG Accuracy	LG AUC Accuracy	NB Accuracy	NB AUC Accuracy
overall_accuracy	0.9984871	0.9999817	0.9931921	0.995854

	NB (Kernel) Accuracy	NB (Kernel) AUC Accuracy
overall_accuracy	0.9432678	0.9973652

Figure 10: Overall Accuracies (Confusion table and AUC accuracies) for supervised algorithms: Logistic regression and Naïve Bayes.



	LG Forloop Mean Accuracy	LG Forloop Variance	NB Forloop Mean Accuracy
overall_accuracy	0.9984115	8.20133e-07	0.9857035

	NB Forloop Variance	NB Forloop with Kernel Mean Accuracy
overall_accuracy	2.879366e-05	0.9305598

	NB Forloop with Kernel Variance
overall_accuracy	0.0002447429

Figure 11: Forloop confusion table accuracies mean and variance for supervised algorithms: Logistic regression and Naïve Bayes.

From figure 10's summary of the prediction accuracies (confusion table accuracy and AUC accuracy) for the supervised machine learning algorithms; Logistic regression and Niave Bayes (with and without the use of the kernel function), we can make the following statements: Logistic regression has an accuracy of 99.84871% = 99.85% and an AUC accuracy of 99.99817% = 100%, Naive Bayes without using kernel has a 99.31921% = 99.32% and an AUC accuracy of 99.5854% = 99.59%. As expected, using Naive Bayes with the kernel, we altered the performance/ accuracy of the test where the accuracy was 94.32678% = 94.33% and an AUC accuracy of 99.73652% = 99.74%. (See RMarkdown Part 8 for raw code). As mentioned, Accuracy and AUC are two types of evaluation metrics to measure the performance of the model. Both are helpful for comparing and assessing how well a model is doing and concluded, based on the dataset response variables being balanced we chose 'accuracy' as the measure of performance. AUC was provided as a means of comparative differences (which in this case is a very small difference) between the two accuracies but ultimately in this case, further supports our confidence in the performance of the models.

Overall, looking at all the supervised machine learning algorithm performances we have Logistic regression with the best performance (99.85%), followed by Naive Bayes without kernel (99.32%) and Naive Bayes with the kernel (94.33%) with the worst performance. From these results, although all performed very well, it is recommended to use the Logistic Regression supervised machine learning algorithm as a predictive model as it performed with the highest accuracy. For loops were conducted for each of the supervised machine learning algorithms above. Figure 11 depicts the results, and we can conclude mean accuracies of 99.84%: Logistic regression, 98.57%: Naive Bayes without kernel, and 93.05%: Naive Bayes with the kernel. The overall variance for each forloop was < 0.0002447429, which solidifies our confidence in the model's performances. (See RMarkdown Part 9 for raw code).

Unsupervised Machine Learning Algorithm: Agglomerative Hierarchical Clustering (AHC)

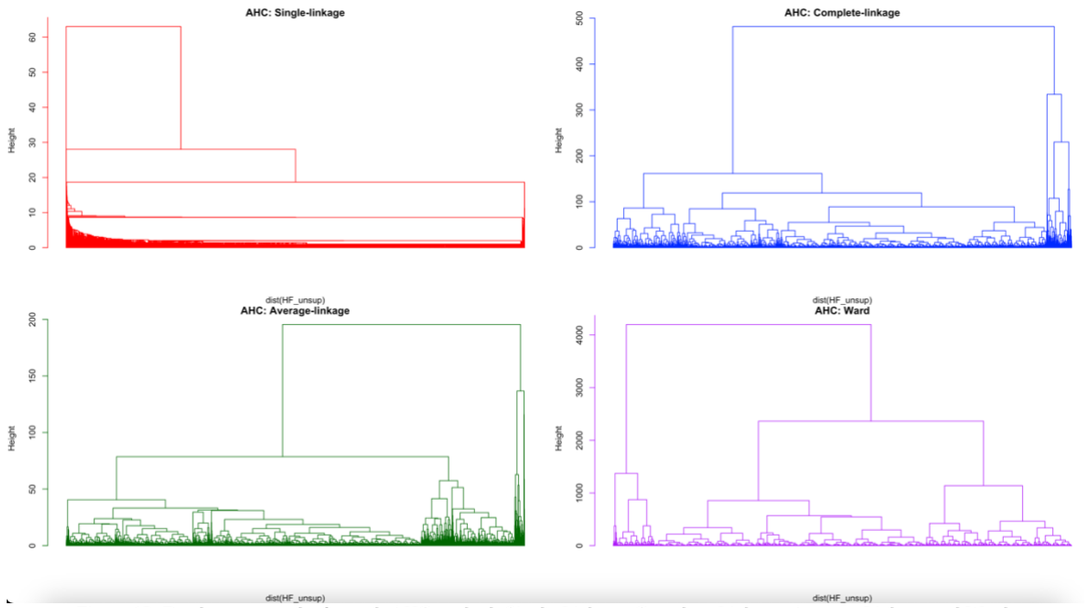


Figure 12: Dendrogram results for each AHC method; Single-Linkage, Complete-Linkage, Average-Linkage and Ward.

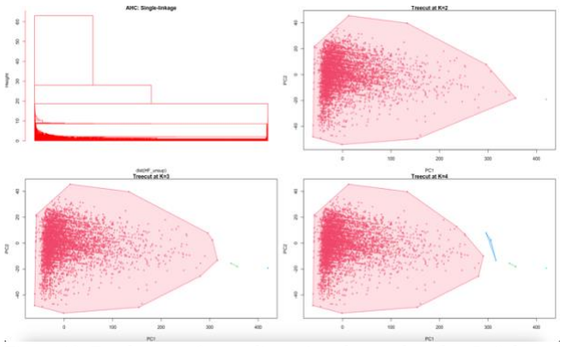


Figure 13: Dendrogram & two-dimension cluster plots (k = 2-4) for Single-Linkage.

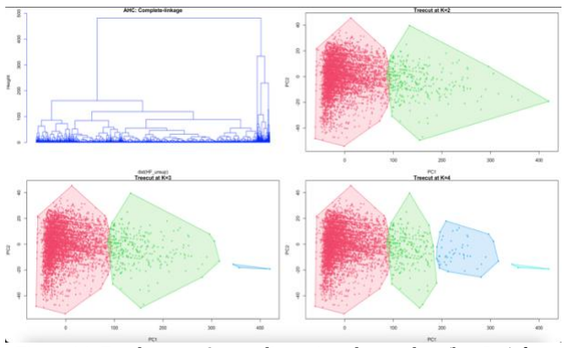


Figure 14: Dendrogram & two-dimension cluster plots (k = 2-4) for Complete-Linkage.

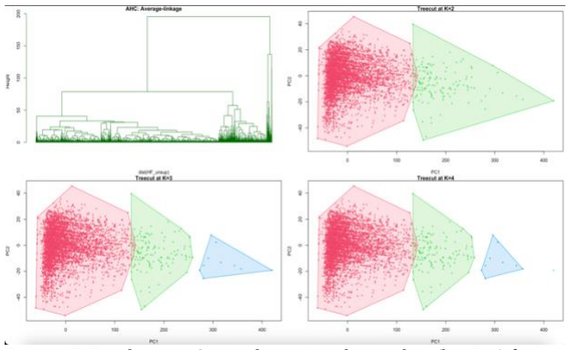


Figure 15: Dendrogram & two-dimension cluster plots (k = 2-4) for Average-Linkage.

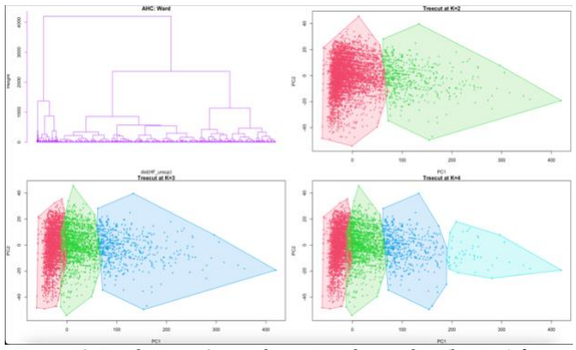


Figure 16: Dendrogram & two-dimension cluster plots (k = 2-4) for Ward.

With unsupervised machine learning algorithms, we are only given input objects without any correctly identified outputs. The goal of unsupervised learning is to discover interesting patterns in the data, which is called knowledge discovery. For this reason, we take the data subset and remove the response variable 'HeartFail' and use the other inputs to learn and possibly discover interesting new patterns/ clusters in the data (See RMarkdown Part 11 for raw code).

With the Agglomerative Hierarchical Cluster (AHC) there are four methods; 1) Single-linkage algorithm (SL): the distance between two clusters of observations, is defined as the smallest pairwise distance between these clusters. 2) Complete-linkage algorithm (CL): a measure of dissimilarity between two clusters, where we take the largest distance. 3) Average-linkage algorithm (AL): an average measure of the pairwise distances between two clusters and finally 4) Wards: aggregating two groups so that within-group inertia increases as little as possible to keep the clusters homogeneous (attempts to generate clusters to minimize the within-cluster variance). Each method was conducted and investigated to observe which had the best performance and within that, at which clustering size provided the most distinct clusters. The results of each method dendrogram are found in figure 12 and figure 13-16 depict individual methods dendrogram and treecuts made clusters sizes 2-4 displayed on a two-dimensional plane to observe

and overall determine which method provides the best hierarchical clustering results and for which clustering size does the method provide the most distinct clustering.

Figure 12 provides the dendrogram (Tree plot) results for each AHC clustering method where the bottom (leaves) are the observations and clusters are represented by the vertical lines. The horizontal lines depict the mergers of the clusters which occur at different levels (heights) on the y-axis scale; corresponding to the distance between the clusters being merged. As mentioned previously the clustering starts with each observation as its own cluster and the two closest points then merge to form a new, larger cluster and then picks the next closest cluster (based on distance) and joins them. This iterates until all clusters have been merged into one cluster. At lower heights (depicted on the y-axis) where the merging of clusters begins, this correlates to greater similarity in the groups of observations. Vice-versa, the greater the height, the greater the difference in the groups of observations. We can cut the dendrogram at any height and/ or cluster size using cuttree () to view the clusters at that specific point in the dendrogram on a two-dimensional plane.

Observing each of the dendrograms we can conclude that SL (figure 12: red dendrogram) performs the worst in creating distinct clusters as several linkages occur at close distances (heights) with no prominent clusters far away from each other relative to their internal distances. We can see in the two-dimensional plots (figure 13) at k = 2-4 cluster sizes there is one distinct disproportional (to the other clusters) cluster, and the other clusters have only one observation (k=2-4), two observations (k=3&4) and 3 observations (k=4) in their identified clusters. Although you cannot see this clustering separation distinctly occurring on the dendrogram, the small cluster sizes would be represented on the far left of the dendrogram when a straight vertical line would span the whole y-axis scale (one observation as a cluster for k=2), another merge near the bottom with two observations (forming a cluster of two for k=3) and, at k=4, another merging to form the fourth cluster containing the three observations.

Investigating CL and AL (figures 12: blue and green dendrograms) we can conclude they both methods produce approximately the same results with a trade-off in clustering performances (figure 14: CL and figure 15: AL) at k=3 and k=4 when observing their two-dimensional plots. Both overall show better results than the SL method; better clustering performance as the cluster linkages occur at further distances with more prominent clusters than SL and the clusters are further away from each other relative to their internal distances. With that said, we can see the scaling for height regarding the CL method having a larger parameter (figure 12: 1-500 vs 0-200 in AL) which correlates to CL's method cluster linkages occurring at further distances with slightly more prominent clusters. This correlates to the two-dimensional plots cluster (figure 15) being more balanced regarding the number of observations, particularly pertaining to the two distinctly larger clusters, with arguably better clustering at K=4 as well. At k=3 it is apparent that AL (figure 15) has done a better job in clustering than CL.

Figure 12: purple dendrogram, depicts Ward's method which produced the best results as we observe both the dendrogram and the two-dimensional plots (figure 16) depicting the clusters at different 'k' values. The Wards dendrograms depict the best clustering performance as the cluster linkages occur at far greater distances; observing the height scale we can see the parameter for the height is from 0 to > 4000 compared to the other methods where their parameters are in the hundreds. We can also observe far greater prominent clusters compared to the other methods as the clusters are further away from each other relative to their internal distances. Referring to the two-dimensional plots, we can see for each cluster value (k=2-4), the wards method provides more prominent clusters as each cluster contains enough observations to consider them significant groups. Investigating further into the clusters we can determine that k=2/ h = 300 provides the best clustering as there are two distinct clusters of observations that are fairly distanced from each other (very little overlap) and enough observations within each cluster to consider both significant groups. With that said both clustering at k=3 and k=4 perform well (compared to the other AHC methods) with the main distinction between k=2 amnd k=3 and 4 is how well k=2 clusters regarding the amount of overlap occurs between the red and green clusters. This distinguishes k=2 as a better clustering performance than at k=3&4. With that said, it may be worth investigating the possibility of having more than two possible outcome groups in the response variables (Heart Failure) due to the natural clustering resulting from the ward's method.

Regarding the dendrograms in figure 12 and the correlation to heights and when cluster linkages occur, due to the varying heights and the compactness at the 'leaves' of the dendrogram it is difficult to determine when linkages begin for comparing between different methods to determine which method starts linkages at an earlier height to determine which is best at finding similarities between observations. For the reason mentioned above, we cannot make any concrete conclusion regarding this aspect of the AHC discussion. Overall, using all AHC methods and observing their results in figures 12 -16, Ward's clustering has the best clustering ability (performance) with the best k-value for most optimal/ distinct clustering being k=2, which (as we would expect) does correlate to the outcomes in the response variable in the original dataset. (See RMarkdown Part 12 for raw code).

Unsupervised Machine Learning Algorithm: DBSCAN and HDBSCAN

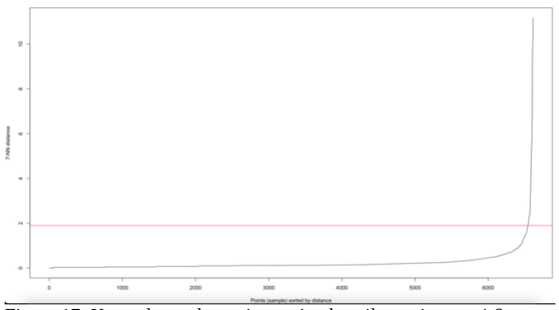


Figure 17: Knee plot to determine optimal epsilon point: $\varepsilon = 1.9$

```
DBSCAN clustering for 6611 objects.
Parameters: eps = 1.9, minPts = 8
Using euclidean distances and borderpoints = TRUE
The clustering contains 10 cluster(s) and 49 noise points.

   0    1    2    3    4    5    6    7    8    9   10
  49 1471 3113 1461  113   52  186   39   25   46   56

Available fields: cluster, eps, minPts, dist, borderPoints
```

Figure 18: Summary of DBSCAN clustering performance.

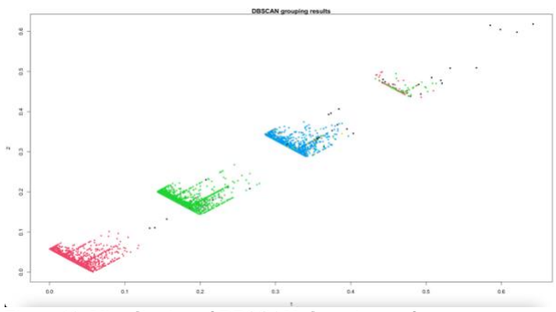


Figure 19: Visualisation of DBSCAN clustering performance.

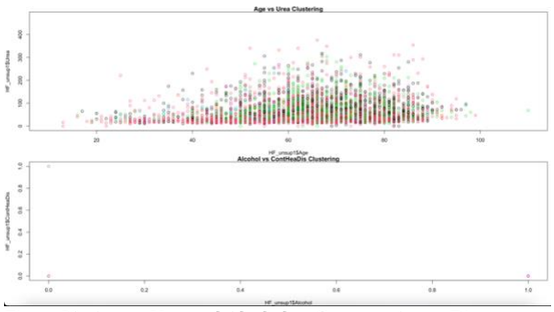


Figure 20: Age vs Urea and Alcohol vs Coronary Artery Disease with DBSCAN results as the colouring function.

As previously mentioned, to use DBSCAN and HDBSCAN the dataset needs to have numerical inputs which is not satisfied with the original and subset data. As a result of this, using the daisy() function, we computed a Gower distance matrix (which contains the Gower distance between points from the mixed variables in the dataset) that can replace the subset dataset in the DBSCAN algorithm. Figure 17 depicts the knee plot which is used to determine the epsilon (eps) value for the DBSCAN algorithm where the optimal value is the point of the graph where the bend/ "knee' occurs. Based on the results we concluded that eps = 1.9 is the optimal value. Another value needed for the algorithm is the minPts value, which, as a rule of thumb, is the number of variables in the dataset. As we know there are 8 variables in the data subset and is used as the minPts value. Overall DBSCAN found a minimum of 10 clusters and 49 noise points. This is a result of trial and error with trying multiple eps and minPts values, knowing the ideal cluster and trying to produce close to the correct number of clusters. Overall, the correct clustering should have found 2 groups, possibly even three with distinct clustering with minimal overlap of clusters. As depicted in figures 19 and 20 there is noticeably a lot of overlap in the clusters and we can conclude from the summary and the visualization that DBSCAN did not perform well and should not be used. When running the analysis in RStudio for the HDBSCAN method, the program ran but never converges to a finished product; 45 minutes running with no results/ finish. This can be a result of the nature of the dataset, specifically the size of the Gower distance object for which the algorithm continuously iterates through with with no final product (See RMarkdown Part 13 for raw code). Overall, based on both unsupervised machine learning algorithms, their methods, and results, we can confidently conclude that AHC is a preferred unsupervised algorithm with the Ward method producing the best results with the most optimal/ distinct clustering being k=2, which correlates to (and what we would expect) the outcomes in the response variable in the original dataset (See RMarkdown Part 13 for raw code).

Descriptive Machine Learning Algorithm: Association Rule Mining

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen
        0.5    0.1    1 none FALSE            TRUE       5    0.48      2     17
 target  ext
  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 3173

set item appearances ...[1 item(s)] done [0.00s].
set transactions ...[50 item(s), 6611 transaction(s)] done [0.02s].
sorting and recoding items ... [26 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 done [4.57s].
writing ... [0 rule(s)] done [0.03s].
creating S4 object  ... done [0.03s].
```

Figure 21: Association Rule Mining Apriori function final summary.

The idea of Association Rule Mining (ARM) involves using machine learning models to iterate and observe the data for patterns, or coincidences, where it classifies reoccurring 'if-then' associations, which within themselves are the association rules. An association rule has two parts: an '(A)ntecedent' (if/left-hand side; lhs) and a '(C)onsequent' (then/right-hand side; rhs). An antecedent is an item found within the data. A consequent is an item found in combination with the antecedent. Therefore, pertaining to this dataset, the intended outcome of using the association rule mining was to find a list of established rules where the consequent is having heart failure and the antecedents are other possible variables within the dataset that correlate to other diseases. The desired outcome is the hope of helping patients and doctors become aware of other possible medical conditions a patient may have that can be tested for and/ or be made aware of if they had heart failure. Unfortunately, through many iterations of different values within the machine learning models 'apriori()' features ('maxlen': maximum number of items in a frequent itemset or rule to be returned, 'support': percentage of cases that include both 'A' and 'C', and 'confidence': percentage of cases with 'A' that also has 'C') 0 association rules were found that pertained to having heart failure as depicted in figure 21 summary. With that said, if more data is collected in the future where more observations and possibly more variables can increase the chances of associated rules being created (See RMarkdown Part 14 for raw code).

VI. Conclusion

Heart Failure has been categorised as a global pandemic where millions of global citizens are affected by it personally, either living with this medical condition or knowing someone who is dealing with it or even worse, someone who has died from it, with numbers projecting to increase in the future. This has resulted in a global burden as it places great stress on not only the people living with it, their families, and friends, but also caregivers and the healthcare system. The projected increase by such an alarming rate can be contributed to the lack of awareness or motivation as a society. To combat this, we need to improve in providing better awareness, incentivising the promotion of self-care through a healthy lifestyle, and continuing to conduct research to provide information to the public and combat this global issue that claims so many lives. Motivated by the need to help and the impacts heart failure has had in my life, this report contributes to the ongoing research and studies being done to provide insights into the characteristics and prediction of heart failure using machine learning algorithms in data mining.

In this report, using the 'DataClean-fullage' dataset, we discussed and compared appropriate supervised machine learning classification algorithms to predict future outcomes of heart failure, conducted knowledge discoveries of the response variable (heart failure) in unsupervised machine learning algorithms to discover possible unknown insightful patterns using the data provided, and used descriptive machine learning algorithms to discover possible associative rules between heart failure other medical conditions represented as other variables in the dataset. The supervised machine learning classification initially resulted in the creation of a subset of variables from the original dataset resulting in the best-performing model; a model that lowers prediction error and finds significant variables using stepwise regression. Using this subset of data, logistic regression, and Naïve Bayes (with and without the use of kernel) were trained and tested in their abilities to predict classification for future observations. It was found that logistic regression had the best performance by producing a testing accuracy of 99.85%, concluding that logistic regression is the recommended supervised classification model for predicting future observations. Our study into the unsupervised machine learning algorithms; AHC and DBSCAN provided an investigation into the process of natural clustering methods where the AHC Ward method provided the best results with the most optimal/ distinct clustering at a cluster size of k=2, aligning with expectation since we already know the outcome of the response variable is also 2. With that said there are some interesting results into the clustering at k=3&4 that may be worth further investigating. Finally, we used association rule mining to investigate the possible association between heart failure and other possible health conditions present in the dataset, with no association rules found. Overall, the power of data provides researchers, scientists, data scientists and individual curious minds to achieve results that may bring us one step closer to improving our way of life. With more data comes more opportunities to get one step closer to solving problems such as the impact of heart failure. This report and its finding do this by providing a possible starting point for further investigating into finding association rules between heart failure and other medical diseases, exploring natural cluster analysis that goes beyond the scope of the expected outcome and providing an effective predictive model to classify heart failure of future patients (observations) that will ultimately combat this global issue that claims so many lives.

VII. References

1. Ponikowski, P., Anker, S. D., AlHabib, K. F., Cowie, M. R., Force, T. L., Hu, S., … Filippatos, G. (2014). Heart failure: preventing disease and death worldwide. *ESC Heart Failure*, *1*(1), 4–25. Retrieved from https://doi.org/10.1002/ehf2.12005 on December 12th.

2. *Coronary Artery Disease - Analysis*. (n.d.). Www.kaggle.com. Retrieved from https://www.kaggle.com/datasets/homelysmile/datacad?select=DataClean-fullage.csv on November 17th 2022.

3. Healthdirect Australia. (2020, September 15). Heart failure. Retrieved from https://www.healthdirect.gov.au/heart-failure on December 9th.

4. CDC. (2019). Heart disease risk factors. Retrieved from https://www.cdc.gov/heartdisease/risk_factors.htm on December 10th 2022.

5. Savarese, G., & Lund, L. H. (2017). Global Public Health Burden of Heart Failure. *Cardiac Failure Review*, *03*(01), Retrieved from https://doi.org/10.15420/cfr.2016:25:2 on December 11th 2022.

6. Mozaffarian, D., Benjamin, E. J., Go, A. S., Arnett, D. K., Blaha, M. J., Cushman, M., Das, S. R., de Ferranti, S., Després, J.-P., Fullerton, H. J., Howard, V. J., Huffman, M. D., Isasi, C. R., Jiménez, M. C., Judd, S. E., Kissela, B. M., Lichtman, J. H., Lisabeth, L. D., Liu, S., & Mackey, R. H. (2016). Heart Disease and Stroke Statistics—2016 Update. Circulation, 133(4). Retrieved from https://doi.org/10.1161/cir.0000000000000350 on December 12th 2022.

7. Wikipedia Contributors. (2019, September 24). Kaggle. Retrieved from https://en.wikipedia.org/wiki/Kaggle on December 13th 2022.

8. Hospital Admissions Data. (n.d.). Retrieved December 13, 2022, from www.kaggle.com website: https://www.kaggle.com/datasets/ashishsahani/hospital-admissions-data?select=HDHI+Admission+data.csv on November 17th 2022.

9. I Webb, G. (2017). (G. I Webb, Ed.) [Review of *Naïve Bayes*]. Retrieved from https://www.researchgate.net/profile/Geoffrey-Webb/publication/306313918_Naive_Bayes/links/5cab15724585157bd32a75b6/Naive-Bayes.pdf on December 12th 2022.