cct College Dublin
Computing • IT • Business

Level 8 Higher Diploma in Data Analytics for Business

***Module**: Strategic Thinking*

Final Report

***Lecturers**:*

Mark Morissey

Graham Glanville

***Student**:*

Dean Watters / SBA20144

Contents Page

Section A

Modelling Technique

According to Hilbe (2009) a Logistic Regression model is a popular machine learning algorithm due to its simplicity in comparison to other algorithms and its nature to provide probabilities and classifying new samples using both continuous and discrete measurements.

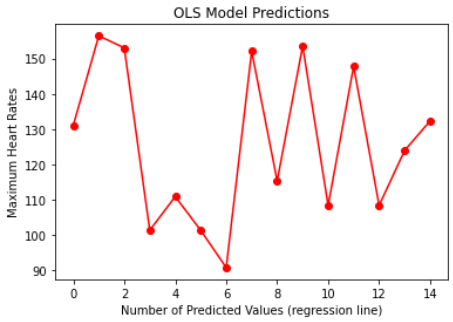
Logistic Regression makes predictions on whether something is True or False (dichotomous outcome) in comparison to Linear Regression which makes predictions on something that has a continuous output. Instead of fitting a line to the data, logistic regression fits an '*S Shaped - Logistic Function*'. This fitted curve ranges from 0 to 1 in which the curve indicates the probability of an event occurring substituted as 0 or 1. Similarly to Linear Regression, a Logistic Regression can work with continuous and discrete (encoded) data however as discussed, the output will be continuous numerical values (Park (2019)).

The following approach has been implemented to break down large datasets into smaller scale datasets to analyse, as it can be difficult to understand, interpret and make statistical inferences from datasets which include many features and scales of large magnitudes.

Therefore, to increase understanding of the key differences between logistic and linear regression models attached is a Jupyter Notebook (see Jupyter Notebook Regression Comparative Inherent Mechanics and Output Analysis). Built is a small-scale data-frame with features with similar characteristics to the heart disease dataset for this project. Prepared data has been inputted into a Linear (Only Least Square Model) and a Logistic Regression Model. Results achieved can compared dissimilarly to get insight into model techniques, processes, and outputs.

For the linear regression model which worked with approximately 4 independent features and 30 observations to make predictions on maximum heart rate. The maximum heart rate dependent variable ranged from a min value of 79.0 to a max value of 194.0 and a mean value is 138.6. The following prediction which can be seen figure (1.0) inferred from the 4 independent coefficients.

Figure (1.0)



As we can see using the four coefficients, they have not effectively been able to predict maximum heart rate within contextual purposes. Considering the context of predicting maximum heart rate levels, in which medical and health professional we can approximate would not find this information helpful due to the trend lines volatility. What would be helpful is an exponential increasing or decreasing trend line considering the four coefficients.

For the Logistic regression model which worked also with 4 independent features and 30 observations mimicking the linear model's independent features to make a prediction. Within the dependent feature lies the key inherent difference. The logistic model will make a prediction on a binary feature (dichotomous outcome) rather than a continuous numeric prediction. We have chosen the above target variable (Maximum Heart Rate) also used for the linear regression model as

the target variable. However, we need to perform some feature engineering on the variable to bin the continuous data creating a binary (dichotomous) variable for the coming logistic regression model. Analyses once again of the summary statistics revealed that the midpoint of this feature is 130 and the mean value 138. This helped determine which data will be collapsed (feature engineered) into which category. Anything greater than the midpoint will proceed into category 1 (> 130) labelled as high heart rate, and all other will proceed into category 0 (< = 130) labelled as low heart rate.

The prepared data was input into the Logistic regression model. The following results were achieved.

Confusion Matrix:

4	7
2	2

Classification Report:

	precision	recall	F1 - score	support
Class 0	0.67	0.36	0.47	11
Class 1	0.22	0.50	0.31	4
accuracy			0.40	15
Macro avg	0.44	0.43	0.39	15
Weight avg	0.55	0.40	0.43	15

We will briefly touch on the confusion matrix as this is what is most interesting to our analysis which is comparing the dissimilarities between linear and logistic regression using more than one quantitative feature.
The confusion matrix represents the raw performance figures of the algorithm of choice. It is composed of the counts of the actual positive and negative class labels on the horizontal axis and those for the predicted class labels on the vertical axis (see Figure (0.0)).

Figure (0.0)

True Positive Count (TP)	*False Positive Count (FP)*
False Negative Count (FN)	*True Negative Count (TN)*

Similarly, both models work with qualitative (encoding required) and quantitative data. Dissimilarly, the key takeaways include that linear and logistic regression require different properties for the dependent variable. The linear model makes a continuous numeric value prediction whereas the logistic regression model makes a prediction on a binary variable (dichotomous outcome). This approach has been helpful to clarify and grasp a better understanding of the inherent characteristics and differences between a linear and logistic regression model.

Logistic Regression Inherent Mathematics and Properties

Osbourne (2015) informs us about the importance of the ability to understand and differentiate between probabilities and odds, when aiming to achieve a better understanding of the internal mechanics of Logistic Regression. From this accurate inference can be made from the output of the Logistic Regression Model.

$$Probability = \frac{Outcomes\ of\ interest}{Total\ number\ of\ all\ possible\ outcomes}$$

According to Park (2019) the probability is the ratio of how likely it is for an event/outcome to occur to the total number of all events/outcomes. Rohatgi & Ehsanes Saleh (2015) further explains that the probability of an event occurring can only go as high as 1, so the probability of something not occurring is 1 – P. The odds is the ratio of something happening to something not happening. Below we can clearly see the formula to calculate the odds of an event occurring.

$$Odds = \frac{P\ (Probability)}{P - 1}$$

The odds ratio is a ratio of two odds and this formula is described below.

$$Odds\ Ratio = \frac{\frac{P\ (Probability)}{P - 1}}{\frac{P\ (Probability)}{P - 1}}$$

Mukhopadhyay (2009) states the problem with the above concepts primarily the probability and odds functions, is that asymmetry makes it difficult to compare odds. Calculating the *log(odds)* alleviates this problem. This is achieved using the log(odds) to make everything symmetrical. The log of the ratio of the probabilities is called the logit function and forms the basis for logistic regression.

$$Logit\ Function = \left(\frac{P}{P - 1} \right)$$

Park (2019) informs us that Logistic Regression is a specific type of Generalized Linear Model. Before getting into the underlying mechanics of a logistic regression model, as mentioned above it is important to understand what to expect in the output of the model. As such we will recap on this as seen below. A point on the logit function close to 1 will indicate a high probability of an event occurring whereas a point on the logit function close to 0.5 will indicate an intermediate probability of an event occurring. A point on the logit function line close to 0 will indicate a low probability of an event occurring.

According to Osbourne (2015), with linear regression the values on the y-axis can in theory range to infinity. Comparatively and dissimilarly, this is not the case for logistic regression, in which the y-axis values are restricted ranging from 0 to 1. To solve this issue in logistic regression the y-axis is transformed from the probability to the *log (odds of (Y))*, so similarly to the linear regression the y-axis can range from – infinity to + infinity. See Logistic Regression function below:

$$Ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k$$

P in this case is the probability of an event occurring and corresponds to a value on the old y-axis between 0 and 1. The midpoint on the old y-axis corresponds to p = 0.5.

An important consideration to note is that even though we associate the graph with 'S' shaped curve with logistic regression, the *coefficients* are in fact presented in terms of the log (odds graph). Alike linear regression, the best fitted line will include a y-axis intercept and a slope.

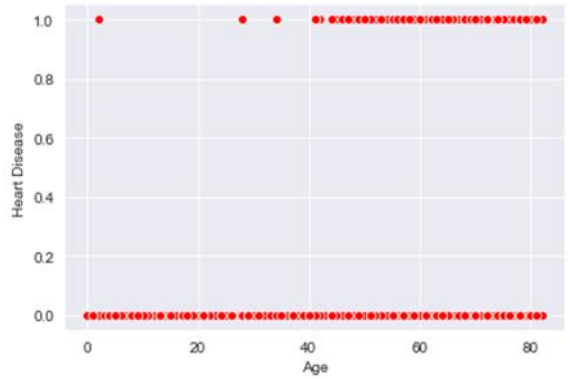
Logistic Regression Modelling Assumptions

Identified through external research were several model assumptions that must be considered before fitting a logistic regression model to the specific dataset (Statistics Simplified (2020)). This research aligned with practical experience and exposure to logistic regression modelling throughout this year allowed for the following investigation to take place. We will briefly discuss these assumptions whilst providing evidence to determine whether the data meets these assumptions through statistical testing techniques (see Jupyter Notebook Statistical Feature Analysis and Testing).

Model Assumption One

Logistic Regression assumes that the dependent variable is binary (dichotomous outcome) which simply means the feature only takes on two possible outcomes. The dependent heart disease variable within the dataset displays a value of one for heart disease in comparison to zero for no heart disease. Figure (2.0) scatterplot further confirms the variable in question on the y – axis is a dichotomous variable meaning the axis is divided into two parts. This is an important characteristic of the specific independent variable influencing the regression model of choice to be the logistic regression model.

Figure (2.0)



Model Assumption Two

Logistic Regression assumes that the observations present in the dataset are independent of each other. This essentially means observation should not be related or come from repeated measurements of the same observation. We ran into technicalities in this assessment finding it difficult to differentiate between model assumption two and model assumption three.

Model Assumption Three

The logistic regression model presumes there is no severe *multicollinearity* present among the independent variables. Correlation refers to the linear relation between two variables as discussed previously in the report. Collinearity refers to an issue when running a regression model when two or more independent variables have a strong linear relationship. Multicollinearity is a special case of collinearity referring to this issue where a strong linear relationship exists between three or more independent variables even if no pair of variables has a high correlation in which unique and autonomous information is not provided to the logistic regression algorithm. This may affect the model if the degree of correlation is high enough between independent variables causing issues for fitting and interpreting the model.

As discussed previously in the Data Cleaning Report the pearson's correlation coefficient test was completed (see figure 2.1) eliminating the dilemma of multicollinearity being present in our data

among the numerical features. The threshold value set for the pearson's correlation coefficient test was (<= 0.7 />= 0.7) was not surpassed as highlighted previously within the data cleaning report (see page 30).

Figure (2.1)

	Age	Average Glucose Level	Body Mass Index
Age	1.000000	0.238060	0.326284
Average Glucose Level	0.238060	1.000000	0.168767
Body Mass Index	0.326284	0.168767	1.000000

Therefore, we can approximate the logistic regression model will fit well to data as the dilemma of multicollinearity has been out ruled.

However, as we aim to train the model with a dataset consisting mostly of categorical features an investigation was undertaken aiming to find out how to test categorical variables for *co – occurrence*. This term simply refers to categories carrying the same statistical information as other. We identified a hypothesis test known as the chi square statistical test. According to Voinov et al. (2013) this test is used to determine whether there may be a statistically significant difference between expected and observed frequencies in one or more categories of a contingency table. The results from the chi square statistical test revealed that among categorical variables there was strong co-occurrence existing. Figure (2.2) below provides a sample of one of the tests for the Marriage Status feature.

Figure (2.2)

```
Marriage Status

stat=6.559, p=0.038
Gender and Marriage Status Probable Dependent

stat=136.683, p=0.000
Marriage Status and Hypertension Probable Dependent

stat=1644.109, p=0.000
Marriage Status and Work Type Probable Dependent

stat=0.175, p=0.676
Marriage Status and Residence Type Probable Independent

stat=599.046, p=0.000
Marriage Status and Smoking Status Probable Dependent

stat=58.924, p=0.000
Marriage Status and Stroke Probable Dependent
```

For further results (see Jupyter Notebook Statistical Feature Analysis and Testing) and accompanying (see Dataset Feature Statistical Testing Report: page 02 - 05). Due to time constraints restricting external research to assess this dilemma for the project, a decision was made to test and analyse inherently the affect off training and testing the logistic regression model with co-occurring categorical features. This was an interesting research aspect of the project in which we will aim to use evidence-based research to answer this question in the concluding parts of the project.

Overall, we can state the data meets this model assumption, in which the logistic regression model assumes there is no severe *multicollinearity* present among the independent variables as this has been founded to be true.

Model Assumption Four

Logistic Regression assumes there are no extreme outliers or out of context influential observations existing in the data. According to Pirker (2009) one of the most common methods for detecting outliers is performing the Cook's distance metric for each observation. Alternatively, domain knowledge combined with visual analysis can help manually detect such observations. As we discussed previously in the Data Cleaning Report (see page 32) an investigation into probable outliers out ruled this dilemma. Therefore, we can confidently state that the data meets the logistic regression modelling assumption four.

Model Assumption Five

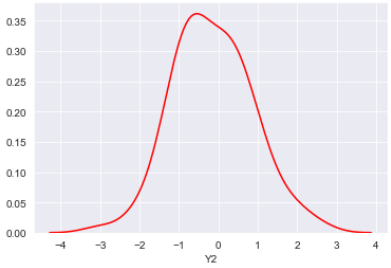
The Logistic Regression Model assumes there is an existing linear relationship between each individual independent variable and the logit function. It is important to understand that this does not mean or require independent and dependent variables to be related linearly as in linear regression modelling. A method for testing whether this assumption is met is a Box – Tidwell test. There is on-going research being conducted for implementing this specific statistical test to assess this model assumption.

Model Assumption Six

The Logistic Regression model assumes that sample size of the dataset is big enough to extract informative and accurate inferences from. We can confirm the data meets model assumption six with 5110 observations and 12 features meeting our dataset size requirements as highlighted previously in the data collection report (see page 08 – 10).

Prior to assessing whether the data has met all the modelling assumptions for the logistic regression model an interesting discovery was made. An investigation and analysis of the quantitative numeric variable's distribution using both the *Shapiro Wilk Test* and scatterplot was undertaken. However, prior to this as the Shapiro Wilk Test is a new statistical testing technique for the project researcher, therefore, to confirm this tests accuracy implemented was a separate testing environment for evaluation. Four features with randomly generated numerical data were developed (see Jupyter Notebook Testing Accuracy of Shapiro - Wilks Test on Randomly Generated Data). One of these features developed through the np.random.normal package which is used to create an array of normally distributed numbers (see Figure (2.3).

Figure (2.3)



The following results were generated (see Figure (2.4)).

Figure (2.4)

```
Test 1
stat=0.922, p=0.000
X1 variable probable not to follow a Gaussian Distribution
Test 4
stat=0.950, p=0.001
```

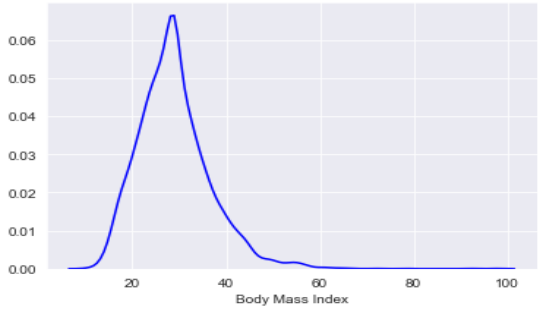
```
Y1 variable probable not to follow a Gaussian Distribution
Test 3
stat=0.954, p=0.001
X2 variable probable not to follow a Gaussian Distribution
Test 4
stat=0.992, p=0.846
Y2 variable probable to follow a Gaussian Distribution
```

The generated results confirm the Shapiro Wilks Test accuracy as were able to correctly confirm that Test 4's distribution is probable to follow a gaussian distribution in contrast to correctly confirm all other distributions are probable not to follow a gaussian distribution. This provided confidence movi ng forward with this specific statistical testing technique.

Utilizing the Shapiro Wilks Test on the heart disease dataset revealed none of the three independent features where normally distributed (see Jupyter Notebook Statistical Feature Analysis and Testing) and accompanying (see Dataset Feature Statistical Testing Report: page 02 - 05). Figure (2.3) display both Shapiro Wilk Test and scatterplot of the Body Mass Index.

Figure (2.3)

```
Body Mass Index
stat=0.952, p=0.000
Body Mass Index variable probable not to follow a Gaussian Distribution
```



Initially, as all three tests and visuals revealed variables within the heart disease dataset were not normally distributed this was seen as a problem for the logistic regression algorithm. However, through research we uncovered one of the key dissimilarities between the linear and logistic regression being the model does not require residuals to be normally distributed or the residuals to have constant variance also known as homoscedasticity (Aguinis (2004)). Furthermore, interestingly uncovered was dissimilarly to linear regression modelling, the logistic regression model does not require a linear relationship between the independent features and the dependent feature.

Concluding modelling assumptions, we can state evidently that we have met four out the six-assumptions discussed above. This has boosted confidence for the machine learning modelling stage in relation to achieving project objectives and goals. Model assumption two and model assumption five remain unclarified in which research is currently being carried out to determine and solve this issue. Furthermore, we will be investigating and testing the logistic regression algorithm's ability to work with categorical variables that have a high degree of co – occurrence among themselves within the modelling phases of the project.

Feature Engineering & Encoding

Due to time constraints in the previous data cleaning stage feature encoding was pushed forward to a later date of the project. This is an important aspect of the project as the logistic regression algorithm requires independent features data to be numerically formatted (Muller & Guido (2017)).

As many independent features from the cleaned data are categorical, approximately 8 we implemented three types of feature encoding for analysis. This included feature encoding the categorical features using one hot encoding and dummy encoding. Label encoding was out-ruled due to the limited number of the features that the model could work with therefore depreciating results to be achieved.

Dummy encoding transformed the original dataset shape increasing the number of features from 12 to approximately 18. One hot encoding transformed the original dataset shape increasing the number of features from 12 to approximately 25. After analysis and consideration of all three approaches (see *Jupyter Notebook Encoding Testing and Analysis*), the decision was made to opt on using the one hot encoding method as it provides greater choice approximately 7 additional features for the algorithm in comparison to the dummy encoding method.

Principal Component Analysis

According to Vidal et al. (2016) principal component analysis is a dimensionality reduction technique frequently utilized to reduce the dimensionality of datasets with many dimensions. On an experimental basis considered was the popular dimensionality reduction technique PCA for inputting our cleaned data to reduce the number of features whilst still withholding most of the information. To investigate and develop a better understanding of this unsupervised dimensionality reduction technique developed was a small – scale data-frame with randomly generated data which has no similar properties to the specific heart disease dataset we are working with for this project (see Jupyter Notebook PCA Dimensionality Reduction Technique Analysis).

Analysed was the affects PCA transformation has on the original randomly generated data by comparing properties of both. Figure (2.4) shows 2/3 features of the original data via scatterplot. We can clearly see the shape and scale of the data from this plot. Figure (2.5) shows the same 2/3 features however instead this after PCA transformation has been complete. There are significant differences in both these plots when we compare scales and shape.

Figure (2.4)

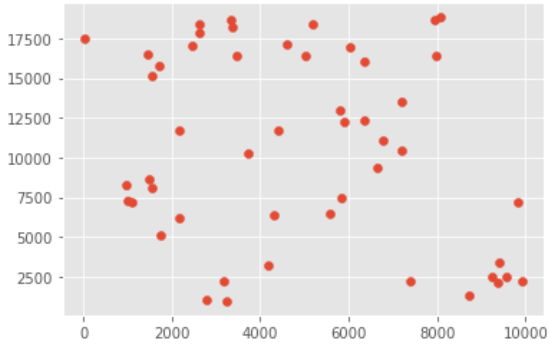
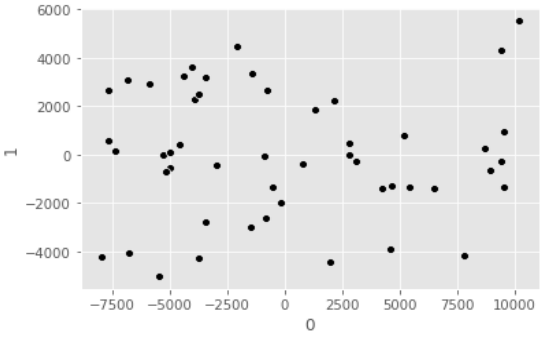


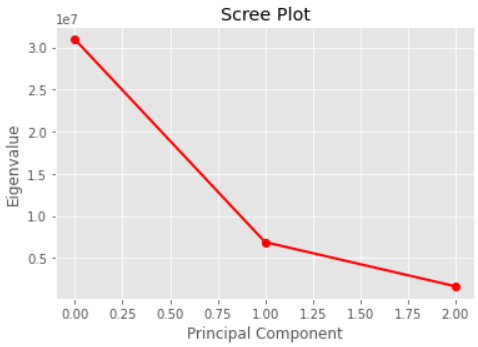
Figure (2.5)



Interestingly figure (2.4) which is principal component one and principal component two captures approximately 95% of the variance from the original three features. Therefore, we can envision how having many features and using this dimensionality reduction technique can still capture most of the data's variance which essentially is information within the data.

Below figure (2.5) shows a plot consisting of the eigenvalues for each component. As mentioned above we can clearly see from this plot that principal component one and two holds most of the data's variance. This is a common usage practice to plot this point for choosing the number of principal components to use.

Figure (2.5)



Concluding our investigation is that to fully understand and grasp the concept of principal component analysis good foundations in mathematics is required specifically regarding eigenvalues and eigenvectors which is an in -depth mathematics topic. PCA has been out ruled based on this as inputting data into an unsupervised dimensionality reduction technique would require time to research further. Furthermore, solidifying this decision on the basis that our data dimensionality does not meet the scale for such a technique.

Furthermore, an interesting question to consider would be when we use such an unsupervised dimensionality reduction technique what the following steps are to extract meaningful information from the data in relation to a domain specific topic. That is outside the scope of this project so we will be closing this investigation at this point on PCA.

Generate Test Design

Developed Test Environment

We will be utilizing Jupyter Notebook Python programming language to set up is the following testing environment utilizing sklearn module for building and testing the logistic regression model on the specific data. The implementation of a structured and procedural plan for training, testing, and evaluating the Logistic Regression Model will be detailed in the following report.

It is important to note that the testing environment has been designed for achieving the best possible outcome for algorithm results aligning with the project objectives and goals. Below is a short recap of project objectives and goals in specific relation to the logistic regression model:

- *Build a machine learning model to analyse and test its performance on the specific domain problem. The model of choice is the Logistic Regression Model algorithm, and the domain specific problem is cardiovascular diseases.*
- *Analyse, interpret, and record performance metrics of the Logistical Regression algorithm providing sufficient evidence whether it may or may not be a good model choice for the development, production, and deployment of a large-scale machine learning model to deliver on cardiovascular disease classification and prediction at a regional and national levels in the Republic of Ireland.*

To successfully achieve the above objectives and goals we have decided to complete six model tests on the prepared data with different applied hyperparameter tuning methods. An analysis and comparison of results will be complete. Outlined in the coming Individual Model Testing Gaols, is specific goals for each individual test. The overall goal is to find the best possible parameters for training and testing the model and therefore enabling our ability to provide sufficient evidence whether the logistic regression model may or may not be a good model choice for the development, production, and deployment of a large-scale machine learning model to deliver on cardiovascular disease classification and prediction at a regional and national levels in the Republic of Ireland.

Testing Procedural Plan

Splitting Data into Train and Testing Partitions

After consideration and research, we have opted with a train test split of 70% training data and 30% testing data. The rationale for this 75:25 ratio split has been approximated as the most efficient numeric split due to the datasets size which is approximately 5110 observations and 12 features as stated in previous reports without feature encoding. With feature encoding applied essentially this split means we will be training our model with 3832 observations and 23 features and testing our model with 1278 observations and 1 feature. This approach will be universal for all five tests.

Modelling Evaluation Strategy

We will be using the logistic regression *accuracy score* as the primary evaluation metric for detailed model comparisons and evaluating all six logistic regression model tests. Accuracy will be the main assessment and evaluation metric for model results. We will also be using other effective evaluation metrics included within the *confusion matrix* and *classification report*.

Gollapaudi (2016) informs us that accuracy is the total number of correctly predicted examples as a fraction of the total number of examples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

As the goal of the project has been altered as outlined earlier within the previous report to the design and development of blue-print national plans rather than a finalized model for the deployment and on this basis our focus will be primarily on the accuracy score. It is important to note that other metrics will also be analysed. Furthermore, it is important to note that in hindsight if plans were proceeding with a finalized model, we would be taking a more in depth and detailed look at other metrics such as balanced accuracy, sensitivity, specificity, precision, and recall.

Furthermore, Gollapaudi (2016) informs us that balanced Accuracy is a good metric if the data under analysis exhibits class imbalance. In this case accuracy may give misleading results.

Sensitivity shows the proportion of positive examples correctly classified in contrast to specificity which shows the proportion of negative examples correctly classified (Mukherjee (2016)).

$$Sensitivity = \frac{TP}{(TP + FN)}$$

$$Specificity = \frac{TN}{(TN + FP)}$$

Mukherjee (2016) inform us that both values fall within the range zero to one. Both these metrics are very application dependent and interestingly in relation to a domain problem such as disease classification they hold increasing value. Essentially the trade-off between both allows us to question whether a percentage of those with a disease being classified as disease free or vice versa is acceptable or not. Ethically in the real world this would not be justifiable however this will be dependent on many factors. For this project we will be keeping a close eye on this.

According to Gollapaudi (2016) precision and recall are closely related to both sensitivity and specificity however the former being particularly applicable to the medical domain in contrast to domains for more informative retrieval. Precision is the proportion of correctly predicted examples that are truly positive. High precision would indicate that only highly likely positives are predicted as

positive.

$$Precision = \frac{TP}{(TP + FP)}$$

Recall is a measure of how complete the results are in which high recall means capturing a large portion of positive examples.

$$Recall = \frac{TP}{(TP + FN)}$$

All results will be recorded (see Model Test Template page 00 - 00) and discussion result variation. We aim to have an optimal performing model at the end of the testing period.

Algorithm Updated Objectives

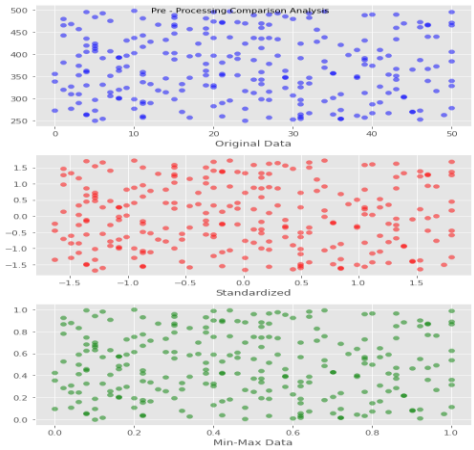
Furthermore, we have identified two primarily new objectives post model technique, assumption, and test generation stages, which were not included in initial objective and goals outlined in previous reports for this project. These include the following and are related to the logistic regression algorithm inherent mechanics.

Firstly, we question how the logistic regression will perform with categorical features encoded numerically in the case were statistical testing previously revealed strong co – occurrence being present among these features (*see Feature Statistical Testing Report*). We will be testing and analysing this dilemma by dropping co-occurring features based on the feature statistical testing report. This will be model test six and we will compare cross sectionally this with our best performing model. In the concluding parts of this project, we aim to provide sufficient evidence on this dilemma.

Secondly, considered was how a logistic regression algorithm performs when the features data being inputted into this algorithm vary significantly in magnitude. To investigate this further developed was a small-scale data-frame with randomly generated data which has no similar properties to the specific heart diseases dataset we are working with for this project (see Jupyter Notebook Testing and Comparing the affects of scaled and unscaled data on the logistic regression model) & (see assisting Report 3).

The data has been both standardized and minmax-scaled using sklearns scaling methods. We inputted the original, standardized and minmax-scaled data into a logistic regression model. Three tests were undertaken following the above scaling techniques in which the data magnitude was increased for each test. Figure (2.6) displays an observational sample visualization of one of the tests scaling transformations.

Figure (2.6)



All three tests showed little to no variation in results when comparing the original, standardized and minmax scaled data. Furthermore, we will be scaling the cleaned heart disease data and inputting this into the logistic regression model performing either a standardising or minmax scaled pre-processing technique.

The aim in the concluding part of this project is to document fact-based evidence on the logistic regressions ability to perform with or without feature co-occurrence. Moreover, we aim to provide sufficient evidence on how logistic regression models perform with data pre-processing scaling techniques. This has been an interesting aspect of the research and aligns with project objectives and goals.

Section B

Build Model

Through research we have identified the following hyperparameter tuning settings associated with the logistic regression model from the sklearn library (Hackeling (2017)). Research conducted has aided in both the identification and proposed value settings for the six tests we aim to carry out. Furthermore, we have developed skills in grid search for the best sets of hyperparameter settings which is implemented within the later model building.

At this point it is noteworthy to mention that a strong mathematical and statistical background is required to implement these hyperparameter settings to a skilled and successive level. Our goal is simply to optimize the model and asses results via result metrics as outlined in the testing procedural plan. Below we will briefly outline the hyperparameter settings we will be utilizing.

Hyperparameter Settings

Penalty

This hyperparameter setting is used to specify which type of normalization the model will implement. The default value for this model is L2 regularization. The values for this parameter setting include L1 regularization, L2 regularization, Elastic-net and None. It is important to note that only certain penalty values will work specific solvers.

Penalty

{'l1', 'l2', 'elasticnet', 'none'}, default='l2'

The 'newton-cg', 'sag' and 'lbfgs' solvers support only l2 penalties. 'elasticnet' is only supported by the 'saga' solver.

If 'none' (not supported by the liblinear solver), no regularization is applied.

C

This hyperparameter setting is used to specify the inverse of regularization strength. This value must include a positive float in which smaller values specify stronger regularization.

C

float, default=1.0

Solver

This hyperparameter setting refers to the algorithm to use in the optimization problem.

Solver *{'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'}, default='lbfgs'.*

For small datasets, 'liblinear' is a good choice, whereas 'sag' and 'saga' are faster for large ones.
For multiclass problems, only 'newton-cg', 'sag', 'saga' and 'lbfgs' handle multinomial loss; 'liblinear' is limited to one-versus-rest schemes.
'newton-cg', 'lbfgs', 'sag' and 'saga' handle L2 or no penalty
'liblinear' and 'saga' also handle L1 penalty
'saga' also supports 'elasticnet' penalty
'liblinear' does not support setting penalty to None

Note:

That 'sag' and 'saga' fast convergence is only guaranteed on features with approximately the same scale (scaled data)

Verbose

Verbose *Verbose: int, default = 0* *For liblinear and lbfgs solcers set verbose to any positive number for verbosity*

Max-iter

This hyperparameter setting refers to the maximum number of iterations taken for the solvers to converge.

Max Iter *max_iterint, default=100* *Maximum number of iterations taken for the solvers to converge*.

Below, we have completed an individual modelling report which outlines the individual goals for each model. This is followed by the Model Tests Report. Each model leads into the next regarding hyperparameter tuning and settings aligned with specific modelling goals (see coming Individual Model Test Goals page 21). Researching the hyperparameter settings and tune grid searching for the logistic regression was a big research piece conducted separately in preparations for this stage of the project. Furthermore, in the ending of this project compiled is a specific report on all models built for testing (see assisting Models Built Report page 14 - 16).

Individual Model Testing Goals

- ***Model Test One***

  Goal: Generate Results

- ***Model Test Two***

  Goal: Outperform Model One

- ***Model Test Three***

  Goal: Outperform Model One & Model Two

- ***Model Test Four***

  Goal: Outperform Model One, Model Two & Model Three

- ***Model Five***

  Goals:

  Choose best performing Model (between model one and four) and input data min-max scaled into this model.

  Compare results and record any variations. Discuss findings in concluding parts of project.

- ***Model Six***

  Goals:

  Choose best performing Model (between model one and four) and drop x number of features identified as co-occurring (refer to feature statistical report).

  Compare results and record any variation. Discuss findings in concluding parts of project.

Model Tests Report

Test:

Model Test One

Date:

01 January 2020

Model Type:

Logistic Regression

***Goal*:**

To build a model with default hyper-parameter settings and successfully train input data and test the model on unseen data.

Hyper-Parameter Tuning:

Default Settings (No – Tuning)

75% Training Data / 25% Test Data

Expected Results:

Successfully inputting data into the model and extracting results.

Results:

Confusion Matrix

	True Positive	False Positive
False Negative	1210	3
True Negative	62	3

Evaluation Metrics

```
Accuracy: 0.9491392801251957
Precision: 0.5
Recall: 0.046153846153846156
```

Evaluation:

Accuracy: Classification rate of 95%, which can be considered as good Accuracy.

Precision: The precision score 0.5% of the model is extremely low indicating that the model when it makes a prediction that it is incorrect 99.5% of the time. In the case of these model predictions, when the LR model predicted that

an individual would develop heart disease, the model has 0.5% success rate in doing so.

Recall: If there are individuals with heart disease who are in the test set this Logistic Regression model can identify it 0.04% of the time.

Action:

Hyper-parameter Tuning for Model Test Two.

Comparing Model Test One & Model Test Two Results.

Test:

Model Test Two

Date:

10 January 2020

Model Type:

Logistic Regression

Goal:

To build a model tuning several hyper-parameter settings and successfully train input data and test the model on unseen data.

To build a model that trains well on the training data as well as the test data.

Hyper-Parameter Tuning:

Model Two

Random State	**31**
Class Weight	**balanced**
Max Iter	**500**
Penalty	**L2**
Solver	**Newton -cg**

75% Training Data / 25% Test Data

Expected Results:

Successfully inputting data into the model and extracting results.

Optimizing results in comparison to previous Model One.

Results:

Confusion Matrix

	True Positive	False Positive
False Negative	922	291
True Negative	14	51

Evaluation Metrics

Accuracy: 0.7613458528951487
Precision: 0.14912280701754385
Recall: 0.7846153846153846

Evaluation:
Accuracy: Classification rate of 76%, which can be considered as not great Accuracy.

Precision: The precision score 1.5% of the model is extremely low indicating that the model when it makes a prediction that it is incorrect 98.5% of the time. In the case of these model predictions, when the LR model predicted that an individual would develop heart disease, the model has 0.5% success rate in doing so.

Recall: If there are individuals with heart disease who are in the test set this Logistic Regression model can identify it 78% of the time.

Model Two and Model One Comparisons

Issue's

Result Interpretation was difficult.

Action:

Hyper-parameter Tuning for Model Test Three.

Data Camp – Hyperparameter Tuning in Python

Implement Cross Validation Technique

Perform Grid-Search to find optimal hyperparameter settings

Comparing Model Test One, Model Test Two and Model Test Three Results.

Test:

Model Test Three

Date:

10 March 2020

Model Type:

Logistic Regression

***Goal*:**

To build a model tuning several hyper-parameter settings and successfully train input data and test the model on unseen data.

To build a model that trains well on the training data as well as the test data.

Hyper-Parameter Tuning:

Model Three

Random state	**31**
Max Iter	500
C	Tune_grid = {'C': np.aramge(0.01,0.99,0.01)}
Cross Validation (CV)	RepeatedKFold(n_splits = 10, n_repeats = 5, random_state = 31)
Solver	liblinear

75% Training Data / 25% Test Data

Expected Results:

Successfully inputting data into the model and extracting results.

Perform a successful Grid Search

Optimizing results in comparison to previous Model One.

Results:

Confusion Matrix

	True Positive	False Positive
False Negative	1209	4
True Negative	62	3

Evaluation Metrics

```
Accuracy: 0.9483568075117371
Precision: 0.42857142857142855
Recall: 0.046153846153846156
```

Evaluation:

Accuracy: Classification rate of 94%, which can be considered as not great Accuracy.

Precision: The precision score 40% of the model is extremely low indicating that the model when it makes a prediction that it is incorrect 40% of the time. In the case of these model predictions, when the LR model predicted that an individual would develop heart disease, the model has 60% success rate in doing so.

Recall: If there are individuals with heart disease who are in the test set this Logistic Regression model can identify it 0.4% of the time.

Model Two and Model One Comparisons

Issue's

Result Interpretation was difficult.

Class Imbalance Identified within target feature

Action:

Hyper-parameter Tuning for Model Test Three.

Data Camp – Hyperparameter Tuning in Python

Implement Cross Validation Technique

Perform Grid-Search to find optimal hyperparameter settings

Comparing Model Test One, Model Test Two and Model Test Three Results.

Test:

Model Test Four

Date:

10 April 2020

Model Type:

Logistic Regression

***Goal*:**

To perform an updated grid search and successfully train input data and test the model on unseen data.

To build a model that trains well on the training data as well as the test data.

Hyper-Parameter Tuning:

Model Four

Random state	**31**
Max Iter	100
C	0.1
Cross Validation (CV)	RepeatedKFold(n_splits = 10, n_repeats = 5, random_state = 31)
Solver	liblinear
refit	True
verbose	3

75% Training Data / 25% Test Data

Expected Results:

Successfully inputting data into the model and extracting results.

Perform an updated successful Grid Search

Optimizing results in comparison to previous Model Three.

Results:

Confusion Matrix

	True Positive	False Positive
False Negative	1211	2
True Negative	63	2

Evaluation Metrics

```
Accuracy: 0.9491392801251957
Precision: 0.5
Recall: 0.03076923076923077
```

Evaluation:

Accuracy: Classification rate of 94%, which can be considered good Accuracy.

Precision: The precision score 50% of the model is an improving result indicating that the model when it makes a prediction that it is incorrect 50% of the time. In the case of these model predictions, when the LR model predicted that an individual would develop heart disease, the model has 50% success rate in doing so.

Recall: If there are individuals with heart disease who are in the test set this Logistic Regression model can identify it 0.3% of the time.

Model Three and Model Two Comparisons

Issue's

Result Interpretation was difficult.

Action:

Choose best performing model built and perform a scaling technique to test whether there are any variations in results. (Test 5)

Evaluate all Model Results comparing and evaluating generated results.

Test:

Model Test Five
(Note: Testing the effects of scaling and inputting the data into the Logistic Regression Algorithm (projects best performing model / Model Two) has on results)

Date:

20 May 2020

Model Type:

Logistic Regression

***Goal*:**

To build a model tuning several hyper-parameter settings and successfully train input data and test the model on unseen data.

To build a model that trains well on the training data as well as the test data.

Data Pre – Processing Technique:

Min Max Scaled

Hyper-Parameter Tuning:

Model Five

Random State	**31**
Class Weight	**balanced**
Max Iter	**500**
Penalty	**L2**
Solver	**Newton -cg**

75% Training Data / 25% Test Data

Expected Results:

We expect there will be no variation in results between Model Test Two and Model Test Five.

Results:

Confusion Matrix

	True Positive	False Positive
False Negative	810	403

```
True Negative     19    46
```

Evaluation Metrics

```
Model Two
Accuracy: 0.7613458528951487
Precision: 0.14912280701754385
Recall: 0.7846153846153846

Model Five
Accuracy: 0.6697965571205008
Precision: 0.10244988864142539
Recall: 0.7076923076923077
```

Evaluation:

Accuracy: Classification rate of 66%, which can be considered as poor Accuracy.

Precision: The precision score 10% of the model is extremely low, indicating that the model when it makes a prediction that it is incorrect 90% of the time. In the case of these model predictions, when the LR model predicted that an individual would develop heart disease, the model has 10% success rate in doing so.

Recall: If there are individuals with heart disease who are in the test set this Logistic Regression model can identify it 70% of the time.

Model Two and Model Five (min max scaled) Comparisons

Difference Metrics

Model Two out - performed Model Five by scoring a better accuracy 10%, precision 5% and recall 8%

Meaning

This provides contradictory evidence in comparison to prior testing on scaling techniques and effects on the Logistic Regression algorithm. This finding will be discussed in the concluding parts of this project.

Issue's

Result Interpretation was difficult.

Class Imbalance

Action:

Drop x number of features identified as co-occurring (refer to feature statistical report). Compare results and record any variation. Discuss findings in concluding parts of project.

Section C

Models Assessment and Evaluation

Model One – Model Four Evaluation

As can be clearly observed (see Model Assessment Report page (06 - 07) all four models returned similar results. Model One, Three and Four returned accuracy scores above 90% which can be seen as a good accuracy score. Model Two interestingly returned an accuracy score of 76% in which we will touch on later in this section.

However, after further research and analysis it was discovered, this metric cannot be solely utilized as an evaluation metric as outlined previously and initially planned. The rationale for this choice we will discuss and to summarize it is in relation to class imbalance.

Analysing collectively all model results we discovered that there was severe class imbalance present within the dataset regarding our target feature. According to Vloymans (2019) class imbalance is present within data when there is a colossal difference when comparing the observations of each class. This posed a massive problem in relation to a domain specific problem such as disease classification. As emphasized earlier in this report there can be minimal error, and this unfortunately contradicts such set project parameters. On such basis accuracy score as initially planned cannot be solely used as an effective metric tool to measure our model's performance.

Moreover, we began to look at other metrics and became particularly interested in the *F1 score* found within all six model classification reports. Research conducted revealed that the F1 metric is a method of combining but precision and recall into a single measurement. The range of the value will be within zero and one, with one representing perfect precision and recall (Sammut & Webb (2011)).

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision \times Recall}$$

Following on from this updated model assessment and evaluation strategy, the concept we are most interested in, is how well our models using the test set is correctly predicts subjects who have heart disease. Analysing Class 1 of all six models classification report we can conclude our models have performed poorly at correctly predicting those who have heart disease within the target feature. As mentioned above interestingly Model Two has performed best at this task returning an F1 score of 0.25. However, this still is evidently a poor result.

Regarding class imbalance approximately 5% of subjects have heart disease in this dataset. We can speculate that due to this class imbalance, our model is not performing well at this task which is predicting those subjects who have heart disease. Concerning the test set which comprises of a support of 65 subjects for our target feature, we are unsure how many of these subjects within each individual test have heart disease. As the target feature holds approximately 5110 subjects in which 276 subjects have heart disease, we can approximate this to be a low value which may add to the model's poor performance in such tasks. We will conclude on this finding in closing stages of this project.

Model Five Evaluation

Model test five which was designed to test if any variation would be identifiable between the projects best performing LR algorithm and a specific pre-processing scaling technique. To investigate this dilemma, model test two was chosen based on the f1 scores returned value for this metric. The returned results show little variation in comparison to model test two. We will cross sectionally compare all evidence we have collected regarding the effects of data pre-processing techniques on the LR algorithm in the closing section of this project.

Model Six Evaluation

Following project feature statistical testing we have decided to proceed with dropping both work type and marriage status for this final test and inputting the data in the projects best performing model (Model Two). The Chi-Square Statistic highlighted both these features as probable for strong contributors to the datasets co-occurrence. The returned results for this final model test showed little to no variation in results when comparing to model test two (see Model Assessment Report page 06 - 07). Previously as discussed within model assumptions three this was an interesting direction for the project in which will discuss in the closing of this project,

Modelling Challenges

Regarding the modelling phase the project ran into some issue's as outlined below. During arising issues, the project researcher remained focused on the project objectives and goals. We will recap on these objectives and goals prior to assessment of data mining results.

Due to lack of experience and mathematic foundations specifically in logistic regression, performing hyperparameter tuning was one of the main challenges in this area of the project. Understanding the inherent parameter implications on the data was difficult and after researching this was evidently more difficult to interpret. The project lacked experience and direction in this area however as already stated in the model building phase, the aim is to find the model that best working with this data. Therefore, focusing on the solution rather than the problem hyperparameter tuning was rationalised and implemented on an experimental basis for all model tests aiming to improve results.

In relation to generated results from all models, issues arose regarding result interpretation followed by associating model results with domain problem (disease classification). The confusion matrix initially posed interpretation issues, however through research this issue was alleviated. Initially regarding model assessment, the project researcher hoped to use one metric for model assessment, however through research and reflection undertaken this strategy is one dimensional and may be missing out on other useful metrics. As outlined previously in the following model assessment and evaluation phase, other useful metrics such as precision, recall and the f1 score were collectively strong model indication metrics.

In total there are 276/5110 subjects suffering from heart disease in this dataset which is approximately just over 5% of the subjects. Therefore 4834 are not suffering from heart disease. There is severe class imbalance present which interferes with model objectives and goals, which is essentially is to classify correctly what subjects are suffering from heart disease and what subjects are not. Furthermore, in relation to the data being split into train and test partitions, the question arises how many subjects are suffering from heart disease who are present in the test set. We can approximate this under the 276 subjects. This links in with the challenge already outlined above regarding result analysis and interpretation. For this domain problem it is imperative that LR model is correctly predicting those subjects (276) who are suffering from heart disease. This is where the most useful information can be extracted from such a model in relation to disease classification. What inherent characteristics are present in those subjects suffering from heart disease. Researching and breaking down the classification report, confusion matrix and other metrics helped evaluate our model's ability to classify those subjects who are suffering from heart disease.

Re – Cap Project Objectives and Goals

Research Objectives & Goals

Align research with dataset analysis and exploration aiming to extract interesting and useful trends in relation to cardiovascular diseases.

Build a machine learning model to analyse and test its performance on the specific domain problem. The model of choice is the Logistic Regression Model algorithm, and the domain specific problem is cardiovascular diseases.

Analyse, interpret, and record performance metrics of the Logistical Regression algorithm providing sufficient evidence whether it may or may not be a good model choice for the development, production, and deployment of a large-scale machine learning model to deliver on cardiovascular disease classification and prediction at a regional and national levels in the Republic of Ireland.

Investigating how the logistic regression will perform with categorical features encoded numerically in which statistical testing previously revealed strong co – occurrence being present among these features (*see Feature Statistical Testing Report*). We will be testing and analysing this dilemma by dropping co-occurring features based on the feature statistical testing report. This will be model test six and we will compare cross sectional this with our best performing model (model one to five). In the concluding parts of this project, we aim to provide sufficient evidence on this dilemma.

Investigate how the logistic regression algorithm performs when the features data being inputted into this algorithm vary significantly in magnitude.

Design and develop blue-print national plans for a large-scale database system focusing on feature selection related to cardiovascular diseases. Findings will be presented to the Health Service Executive for analysis and assessment. This will assist in the reshaping of healthcare provision and the development of an updated strategy focused on preventative rather than curative measures at local, regional, and national levels in relation to cardiovascular diseases.

Conducting research on The Danish digital healthcare model to highlight the impactful and positive results being achieved in Denmark.

Build argumentative evidence highlighting the need for revolutionary changes required to use existing health data universally to better improve healthcare in the Republic of Ireland. We will be using The Open Health Data Governance Strategy 2016 and the Open Data Strategy 2017 -2022 as the foundations for this argument.

Research Fundamentals

Following the CRISP DM Framework to help achieve organized and well formatted research meeting project objectives and gaols.

Complete a good standard of research that is reproducible in the fields of cardiovascular disease and machine learning model research.

Due the contextual background of this project being disease classification completing truthful and ethical research is a crucial foundation influencing credible and reproducible research in the simultaneous fields of machine learning and disease classification research.

An ability to alter, adapt and readapt when required acting within the best interests of project objectives and goals.

Assessment of Data Mining Results

Below we will make a final assessment regarding project succession at meeting specific objectives and goals, whilst aligning with research fundaments. Within the modelling phases the project developed from a more researched based piece regarding heart disease into a more technical investigation into the specific algorithm of choice.

The CRISP DM framework was a crucial tool for supporting and directional guidance throughout the project. We followed all research fundaments successfully following the Crisp DM framework successfully, completing a high standard, ethical and honest research piece contributing simultaneously to the fields of machine learning and cardiovascular disease classification.

The successful extraction of meaningful and useful trends in relation to cardiovascular disease is documented clearly in the Data Exploration Report. This aligned with external research, was the foundations for one of the projects main goals, which was the design and development of blue-print national plans for a large-scale database system focusing on feature selection related to cardiovascular diseases.

Another key goal was to find the best possible parameters for training and testing the model and therefore enabling our ability to provide sufficient evidence whether the logistic regression model may or may not be a good model choice for the development, production, and deployment of a large-scale machine learning model to deliver on cardiovascular disease classification and prediction at a regional and national levels in the Republic of Ireland. Unfortunately, severe class imbalance was a massive issue identified in the later stages of the machine learning modelling phase. After a comprehensive analysis it was discovered that all models performed poorly at classifying those subjects who were categorized as having heart disease. In contrast the model performed well at classifying those who did not have the disease. Therefore, rather than concluding that the Logistic Regression Model does not perform well on this data and would not be a good model choice for this task within the Irish health care system, rather we would emphasize utilizing datasets which class imbalance does not exist in the target feature. Unfortunately, this issue required highlighting early on in such a project but evidently this crucial data mining error was missed.

Based on model test six generated results in relation to the features being inputted into the logistic regression model the evidence collated would suggest based on this specific test that co-occurrence does not affect the logistic regression models performance. Based on these findings we can conclude that co-occurrence does not impact the logistic regressions performance. However, further testing with different datasets would be required to further back these findings. Unfortunately, due to time constraints this was not within the scope of this project however this could have made for an interesting aspect of the project.

Another interesting technical investigation the project took on was how the logistic regression model would perform with data with varying magnitudes being inputted into the algorithm. Results generated were contradictory. The separate testing environment set up with randomly generated data pre-processed showed no variation in results whereas when we inputted the prepared heart disease data (min-max scaled) into our best performing model there was slight depreciation in generated results. Therefore, based on this investigation we can conclude that due to the contradictory evidence founded through cross sectional comparisons further testing would be required with different datasets to justify any conclusional statements.

Furthermore, it is important to mention that a strong mathematical and statistical background is required to implement these hyperparameter settings to a skilled and successive level and to understand the true inherent characteristics and properties of the specific algorithm.

Finally, as outlined previously the project aim was shifting the focus and emphasis from the building of a finalized machine learning disease predicational model for deployment, rather to the design and development of blue-print national plans of a large-scale database system focusing on feature selection related to cardiovascular diseases which will assist in the reshaping of healthcare provision and the development of an updated strategy focusing on preventative rather than curative measures at regional and national levels in relation to cardiovascular disease.

A massive shortcoming of this project relates to our proposed goal above. Unfortunately, it was not foreseen the scale of this specific goal. Firstly, to achieve such goals research would need to be conducted on the most recent and up to date information and data available in relation to Irelands policy and procedures for collecting and sharing health data. Secondly, further research and great knowledge would be required on database systems. Unfortunately, this has not been completed and to make assumptions and suggestions would contradict the research fundamentals set at the beginning of this project.

Nonetheless, the research we have completed cannot be disregarded despite the discrepancies outlined above. This sectional analysis and research completed in relation to the above goal included the identification and analysis of two datasets associated with cardiovascular disease regarding feature proposals which enabled the proposed features for this large-scale database system to be identified and finalized. Expected was that the following features would format the foundations and structure for the proposed large scale health database system to be incorporated within the Irish healthcare system. Unfortunately, further research is required before any further recommendations can be made.

Reference List

Aguinis, H (2004) Regression Analysis for Categorical Moderators. New York: The Guikford Press.

Gollapaudi, S (2016) Practical Machine Learning. UK: Packt Publishing.

Hackeling, G (2017) Mastering Machine Learning with Sciki-Learn. Birmingham: Packt Publishers.

Hilbe, M, J (2009) Logistic Regression Models. USA: CRC Press.

Mukherjee, S (2016) F# for Machine Learning Essentials. UK: Packt Publishing.

Mukhopadhyay, N (2000) Probabilities and Statistical Inferences. USA: Marcel Dekker Inc.

Muller, A, C., Guido, S (2017) Introduction to Machine Learning with Python. USA: O' Reilly Mdeia.

Osbourne, W, J (2015) Best Practices in Logistic Regression. USA: Sage Publications.

Park, A (2019) Python Machine Learning: A Complete Guide for Beginners on Machine Learning and Deep Learning. Italy: Amazon Italia Logistica.

Pirker, C (2009) Statistical Noise or Valuable Information. Germany: Spring Science & Business Media.

Rohatgi, V, K., Ehsanes Saleh, A, K (2015) An Introduction to probabilities and Statistics. Canada: John Wiley and Sons.

Sammut, C ., Webb, G (2011) Encyclopedia of Machine Learning. New York: Springer Publications.

Statistics Simplified / Statology (2020) Available at: https://www.statology.org/assumptions-of-logistic-regression/ (Accessed: 23 April 2020).

Vidal, R., Ma, Y., Shankar Sastry, S (2016) Generalized Principal Component Analysis. USA: Springer – Verlag Publications.

Vloymans, S (2019) Dealing with Imbalanced and Weekly Labelled Data in Machine Learning using Fuzzy and Rough Set Methods. Switzerland: Springer Publications.

Voinov, V., Nikulin, M., Balakrishnan, N (2013) Chi – Squared Goodness of fit Tests with Applications. UK: Elsevier Inc.