Biostatistics (2021), **0**, 0, *pp.* 1–**??**
doi:10.1093/biostatistics/output

December 21, 2021

Creating Value Through Data Mining Project

ROWE Veronica, KOUANGO Malou-Tinette

Geneva School of Economics & Management, University of Geneva, Bd du Pont-d'Arve 40, 1205 Geneva, Switzerland

ABSTRACT

The present research paper analyses a medical data set relating to stroke cases. Its endeavour is to firstly build and compare multiple classification models, extract the explanatory patient's characteristics that most influence stroke occurrences, and finally create a typical stroke-likely patient's profile. Lastly, it discusses what the results imply for data-driven public health campaigns.

1. Introduction

A stroke is a medical emergency that affects the arteries and shortcuts the blood supply in a person's brain, often through the rupture of a blood vessel at that level (Mayo-Clinic, 2021). According to the World Stroke Organization (Organization, 2021), strokes are the leading cause of disability and the second leading cause of death worldwide. It is not presumably linked to any particular gender or genetic condition, and as such, can affect anyone, anywhere, at any time.

In the medical field, data mining, which is the analysis and transformation of data into information through the application of statistical algorithms, has proven very useful in not only

identifying patterns and trends, but also making predictions: data mining helps in detecting patients' risks and/or conditions in advance, and allows for optimized care and treatment.

The present research paper will apply different classification techniques on a stroke cases data set, and will try to determine not the probability of dying from a stroke, but the likelihood to suffer from one depending on a patient's characteristics. We will first visualize (Exploratory Data Analysis), then prepare (Data Pre-processing) and finally apply data mining techniques to respond to three main research questions: Which variables have the highest impact on the liability to get a stroke ? Can we predict the likelihood of a patient getting a stroke with the patient's characteristics ? And lastly, from the data, which strategy/what policies would be the most effective for a public health campaign ?

All theoretical concepts explained in this paper are taken from the book called: "Data Mining for Business Analytics: Concepts, Techniques and Applications in R" (Shmueli and Peter C. Bruce, 2018).

2. Dataset: Exploratory Data Analysis (EDA)

Our data set consists in hospital records of 5110 patients with 12 attributes: ID, gender, age, hypertension, heart_disease, ever_married, work_type, Residence_type, avg_glucose_level, bmi, smoking_status, and stroke (see Appendix for details). Stroke is the response variable, that states if a patient has had a stroke or not.

Our data mining research starts with Exploratory Data Analysis (EDA). The goal of EDA is to get a general picture of our values and variables. By exploring the data first, we are then able to clean and pre-process them. Data cleaning and pre-processing is the act of transforming raw data into well-formatted data sets ready for data mining and model construction. Moreover, by visualizing the distribution of our variables, we can infer preliminary hypothesis based on general patterns.

Fig. 1. Table 1.1: EDA - Initial Data Structure

```
## 'data.frame':    5110 obs. of  12 variables:
##  $ id               : int  9046 51676 31112 60182 1665 56669 53882 10434 2741
9 60491 ...
##  $ gender           : chr  "Male" "Female" "Male" "Female" ...
##  $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
##  $ hypertension     : int  0 0 0 0 1 0 1 0 0 0 ...
##  $ heart_disease    : int  1 0 1 0 0 0 1 0 0 0 ...
##  $ ever_married     : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ work_type        : chr  "Private" "Self-employed" "Private" "Private" ...
##  $ Residence_type   : chr  "Urban" "Rural" "Rural" "Urban" ...
##  $ avg_glucose_level: num  229 202 106 171 174 ...
##  $ bmi              : chr  "36.6" "N/A" "32.5" "34.4" ...
##  $ smoking_status   : chr  "formerly smoked" "never smoked" "never smoked" "s
mokes" ...
##  $ stroke           : int  1 1 1 1 1 1 1 1 1 1 ...
```

We format our data set in R - a software for data science. The categorical variables with type "character" are changed to factors, for appropriate data visualisation and analysis, along with our outcome variable stroke, changed to a 2-levels factor. The variable bmi has "N/A" characters, that we change to real NAs in R and set the variable type to numeric. Lastly, we remove the variable column 'ID', as it is irrelevant to the goal of our research. It only represents the patient identifying number at the hospital and thus has no impact on our outcome variable of interest 'stroke'.

One important point to notice that our Stroke data set is widely unbalanced: the observations happen to be distributed such that 95.127% of patients have not suffered from stroke, while only 4.873% of patients have. The asymmetrical response variable is a salient point we should be wary of. In presence of unbalanced data sets, except collecting more data on the rarer class, one of the most popular solution is resampling the data set: either by oversampling, which is the act of artificially adding minority instances by coping a number of existing instances, or undersampling, the act of removing instances from the majority class to adjust the imbalance (Lahera, 2019). The downside is that both methods create a selection bias: the population may not be as representative

anymore. In our case, since we do not wish to modify the data set, we keep it as it is and will rather use a penalized model, that modifies the classification rule if need be (e.g. with a selected cut-off or uneven missclassification costs).

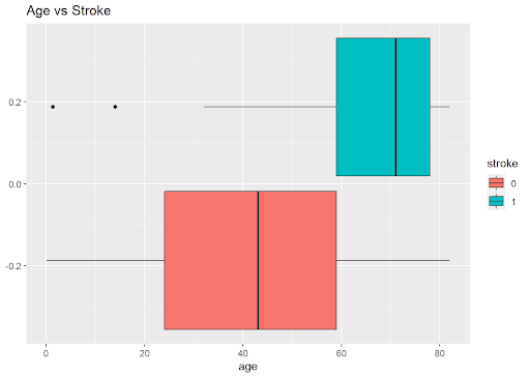
2.1 *Data Visualisation*

In the data visualization part, we plot each variable separately, and explore relationships amongst them; the purpose being to discover patterns and form preliminary hypotheses before going further into the data modelling. By knowing our data first, we can more easily spot errors and/or inconsistencies in our models, and proceed to a more advised selection.

We plot all explanatory variables according to the response variable stroke, by level: we distinguish between stroke and non-stroke, to foresee the variables' dynamics. For our numerical variables, we use histograms and boxplots to learn about the distribution of their values and detect extreme cases (= outliers). Likewise, for categorical variables, we use bar charts (see in Appendix).

Before all else, by simply looking at the data, most observations seem to fall in the range we would expect them to be (no typos or incongruities).

Fig. 2. Graph 2.1: EDA - Age Boxplot



For the variable age, we notice an important difference between both boxplots. The average age of the patient suffering from stroke is 20 years higher than a patient not affected by it. Undeniably, the older the person, the higher their chances of having a stroke. This can be spotted also on the histogram: mostly older patients suffer from stroke. Thus, from EDA alone, we expect age to be a significant explanatory variable.

The variables avg_glucose_level and bmi are right-skewed, and there does not seem to be a significant difference in bmi between stroke and non-stroke patients. As for average glucose level, while non-stroke patients have low glucose level, patients suffering from stroke fall into a wider range. Though medically, a higher Body Mass Index and excessive sugar level tend to indicate a patient's health status, we can not presume their significant influence on stroke simply from our data visualisation.

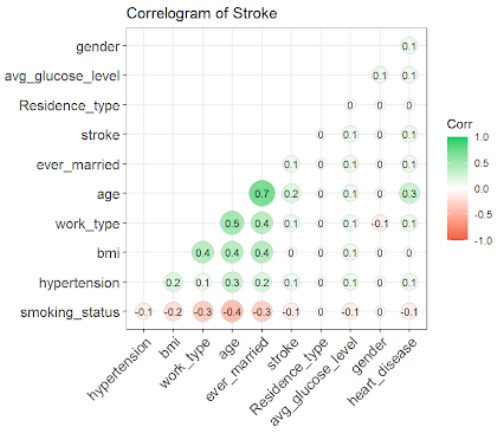
The levels of males and females having strokes are about the same. However, since there are more women in the sample than men, once scaled, it is safe to suppose from the data that men might have higher chances of having a stroke; though the effect/significance level is yet to be determined. In relative terms, a good proportion of patients who suffered a stroke had hypertension. The same applies to heart disease, but to a lower extent. Both variables may, or may not, have a significant effect on stroke. Strokes mostly affect people that have been engaged, since the variable ever_married in its definition must be highly correlated with age: the older a person, the higher the chance they have been engaged. As for the working status, private work type seems to accentuate the likelihood of stroke. However, taking into consideration the higher number of observations of that category, it seems logical that the number of strokes in absolute terms is higher. Residence type is pretty evenly distributed and there seems to be a little difference between rural and urban categories, therefore, we do not expect it to have any significance in the models. As for the smoking_status variable, we can not infer anything from the bar charts. Moreover, the unknown category - which contains a large number of observations - could be causing bias: with such numbers of unknown/NA values, the data set may not be informative enough on the influence of this particular variable.

To summarise, we suspect the "medical terms" (gender, age, hypertension, heart_disease, avg_glucose_level, bmi, smoking_status) to have more influence on stroke than the "social terms" (ever_married, work_type, residence_type). It is commonly known that the general health status

of a person will have an effect on the likelihood of having a stroke, heart attack or other illnesses. However, some social term variables may be correlated with medical ones. Marriage status is the most clear example as older people usually have a higher chance of being married, the same goes for work type with the "children/never_worked" categories connected to a young age. Thus, correlation between variables must be estimated and analysed in order to not assume spurious causes and effects.

2.2 *Correlation Matrices*

Multicollinearity is present when multiple explanatory variables are not independent, but correlated with one another. This leads to redundant/repeated information in regression and classification models, and can cause unreliable estimates on the outcome variable. Moreover, the interdependence of predictors could affect the significance of certain variables on the response, as their full effect is reduced by the correlation, rendering models ill-suited. When possible and if relevant, the number of variables in a data set must be downsized. This is called dimension reduction. For algorithms to be efficient, it is important to remove redundant information in order to not overfit and impact the accuracy of our model. Correlograms help for dimension reduction (variable omission).



We plot the correlation between the variable pairs to learn about possible variables redundancy or interdependence. As expected, our data set suffers from multicollinearity problem between the variables ever_married and age, whose correlation coefficient can be considered high (= 0.7) (M, 2019). Moreover, as we initially thought, the variable work_type is also correlated with age (= 0.5), however not sufficiently to be discarded.

2.3 *Data Preprocessing*

Before further analysing the significance of the explanatory variables with data mining and confirming, or not, our assumptions, we preprocess our data appropriately. For multicollinearity reasons, we have decided to cut the variable ever_married all together from our data modelling. As for the gender variable, it has 3 levels, due to 1 patient identifying as "Other". We decided to simply erase this observation, as it is not representative enough to be taken into consideration. It represents only 0.02% of our data set. Additionally, the variable bmi possesses around 200 N/A values. When the number of missing values is small, they may be omitted from the data set. A recommended alternative is to replace those missing value with the mean: such conversion does not add any information on the effect of the variable on the outcome, but it allows to proceed through the analysis without losing the information contained in the record for other variables. As our data set is already very unbalanced, we did not want to take the risk of omitting those observations, since dropping even a single stroke observation could aggravate further the gap between the number of strokes and non-stroke records. Therefore, to handle missing values, we decided to simply transform them into the mean value of the column.

3. Methodology

Our goal is to develop a forecasting model. For this endeavor, we followed the list of steps recommended for data mining, according to the SEMMA method (Sample, Explore, Modify, Model, Assess). After having explored, cleaned and preprocessed the data, we randomly partition our data set into three parts: training (50%), validation (30%) and test set (20%). The training set is the part of data used for building the model, while the validation set is the part of the data that is used for fitting the model by testing predictions and selecting the best model and adjustments. The test set is the part of data that we will use to assess the predictive power of our final best fitted built model, via what is called Cross-Validation. With our binary outcome

variable 'stroke', we will use five methods of classification.

First, we will run a logistic regression. Logistic regression is a linear regression technique that, unlike other regression techniques, allows prediction for a categorical outcome variable. Using the other variables as predictors, it can be used either to categorize a new unknown record (classification), or to find the differentiating factors between the categories (profiling). The standard equation of the logistic regression model is:

$$\log(odds) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_i x_i. \tag{3.1}$$

with the $\underline{\beta}_i$ being the coefficients of the regression model and x_i being the characteristic variables. In our case, we want our model to be able to do both: classify a new patient as likely or not likely to get a stroke, and identify which circumstances most influence the likelihood to suffer from a stroke. Logistic regression will give us the variable coefficients and their p-value in the model. The variables with a p-value lower than 0.05 will be proven significant in the LR model, while all others will not have a proven effect on the likelihood to get a stroke. We will then find the best cut-off, the one that is able to predict the most strokes with the least misclassifications, and see how it compares to the previous LR with the default classification rule.

Second, we will build a classification tree. As its name suggests, classification tree is a method that categorizes observations into subgroups by splitting predictor variables into branches, following the classification rules This is called "recursive partitioning". It creates a path that observations follow, according to their characteristics, and end up classified in either of the outcome's categories, in our case 'stroke' or 'non-stroke'. In accordance with the most significant variables for CT, we will prune the tree using the validation data. Pruning the classification tree consists in reducing the size of the tree by trading off the number of decision nodes against misclassification errors. The CP (Complexity Parameter) table helps finding the best number of splits in the tree, which is the one that corresponds to the minimum error on the validation data. After pruning the tree, we will use the predictions as propensities (probability), and not binary outcomes (0,1),

to find the best cut-off and see how such classification rule compares to the default CT when predicting strokes.

We will also use the K-Nearest Neighbors (K-nn) technique. The K-nn algorithm computes the euclidean distance between records. The Euclidean distance between two records a,b for each of their n attribute variables i is :

$$euclidean\,(a,b) = \sqrt{\sum_{i=1}^{n} (b_i - a_i)^2} \tag{3.2}$$

K-nn classifies an observation by comparing it to the other observations in the data, finding its k closest neighbors, which are those with the lowest euclidean distance, and attributes the majority class of these neighbor records to the new observation. The best k, which is the number of neighbors used to classify an observation into either class, must be chosen. It will maximize accurate stroke predictions with the least misclassifications on the validation set. Once the best k is found, the optimal cut-off will be set by using predictions as propensities (= probability) instead of binary outcomes (0,1), and the K-nn model with and without cut-off will be compared.

Finally, we will construct two ensembles methods, based on majority vote and average probability. Just like the well-known 'wisdom of the crowd' concept, ensembles methods are based on the simple idea that combining models can help improve predictive power. For each observation, the majority vote ensemble method compares the binary prediction made by each model used and attributes the most voted one by all models to the record. The average ensemble method computes the mean of the probabilities of belonging to a class made by each model used and attributes this average to the record, following an optimal cut-off that we will have to find.

At the end, we will compare all methods used, choose the best one and try to improve its performance by reducing "noise", keeping only the variables that are significant. We will assess the best method's models performances by predicting strokes on the test set (= Cross-Validation), by examining its ROC (Receiver Operating Characteristics) curve, that will show us the two classes

misclassification trade-off. Finally we will analyze the best model's gain/lift and decile charts, that will tell us to what extent does the model perform better than randomness.

As a bonus, we will introduce how Linear Discriminant Analysis (LDA) method could be used in our situation. Similar to Logistic Regression, Linear Discriminant Analysis is a method that is used for both classification and profiling. It computes the distance of a record from each of the class' means, and attributes a record to the class to which it is closest. We will use only the two most significant characteristics to have an insight on how this data mining technique could be used on an unbalanced data set.

4. Results

In the results section of our paper, we use Confusion Matrices to assess the forecasting performance of our different classification models. Before going forward, we define the parameters we are interested in and the specifications to take into account when dealing with a data set from the medical field. Accuracy, sensitivity and specificity are the three main measures of model's predictive power. Sensitivity refers to the rate of true positives, specificity corresponds to the rate of true negatives, while accuracy represents how close the model is to both true measures.

The importance of false positives and false negatives are weighed differently in the medical field than in other industries. In the Sales department, a false negative usually represents someone unexpected who buys the product, resulting in extra profit, while false positives are unoptimal, since resources (money, time, efforts) were used to target a mistaken potential customer. In medical situations, like our stroke outcome, it is the opposite: a false negative symbolises a patient that given our model, should not have a stroke and stays unwatchful about it, but is actually inclined to suffer from one. Such a situation is unwanted, we want a model that minimizes wrongly classifying patients that would have a stroke. In contrast, false positives, that symbolise patients unlikely to have a stroke that are believed inclined to by the model, do not have as

much negative human repercussions than the case above. Though it could be stress inducing to mistakenly expect a stroke, the theoretical goal of data mining in the medical field is to save lives, and this is achieved by balanced appraisal measures.

Thus, since our model's ultimate goal is to be able to predict stroke cases, it must concentrate on the sensitivity measure. However, taking sensitivity as the only assessor of our model's performance would lead to all observations being classified as stroke cases. But having too many false positives is not optimal either, since it is costly to invest in patients inaccurately. So along with sensitivity on one hand, we are also interested in accuracy measure: does the model predict the most potential stroke cases correctly, while staying closest to reality such that it misclassifies the least observations possible? We will use what we call an Accuracy-Sensitivity mix: the trade-off between both measures as the core indicator of our model's performance.

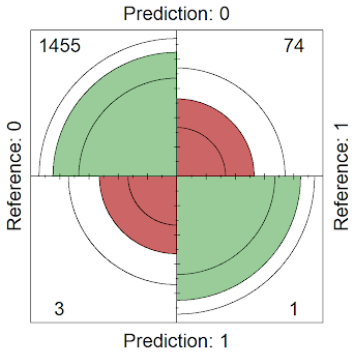
4.1 *Logistic Regression*

We start by running a general logistic regression model (GLM) method on all of the pre-processed variables (see in Appendix: Table 1: LR - Generalized Linear Model).

We have 4 significant variables (in decreasing order of significance): age, hypertension, heart_disease avg_glucose_level. This seems coherent with our data visualisation: these are all medical terms. From a critical point of view, usually medical measures have a higher direct impact on diseases than social ones, especially if these are just facts like "rural" or "urban" instead of relevant pollution measures that could be critically considered. As our preliminary hypothesis stated, one can naturally assume that they would be significant on patients' general health, relating to stroke likelihood. By visualising the significant variables compared to the outcome in relationship graphs (Appendix: LR- Graph 1,2,3,4), we can confirm that the older the patient is the higher the likelihood of having a stroke. This is shown by the line that goes steeper according to a higher age. Moreover, the stroke occurrences are more concentrated on the right side of the graph, indicating

an older age. For both hypertension and heart_disease, we can see that the line goes slightly up on the right side of the relationship graph, symbolising a positive effect (relationship) between stroke and the explanatory variable. As for the average glucose level graph, we can see that correspondent stroke occurrences are more volatile, however, there is a weakly positive relationship with stroke, since the relationship line increases.

Fig. 3. Table 2.2: LR - Initial Confusion matrix



Our initial Confusion Matrix returns very poor results. This is to be expected as we have not adapted the cut-off value yet: it is left at the default value of 0.5. As there are more non-stroke occurrences, it is logical that our model predicts all predictions as the majority class at the default cut-off. Given our previous results and the distribution of our outcome variable, we must adapt our cut-off to fit our data sample. Therefore, we correct this by adjusting the separation percentage. In order to visualise their trade-off, we plot both accuracy and sensitivity on a graph, and find both lines' meeting point at 0.06, which is the optimal cut-off.

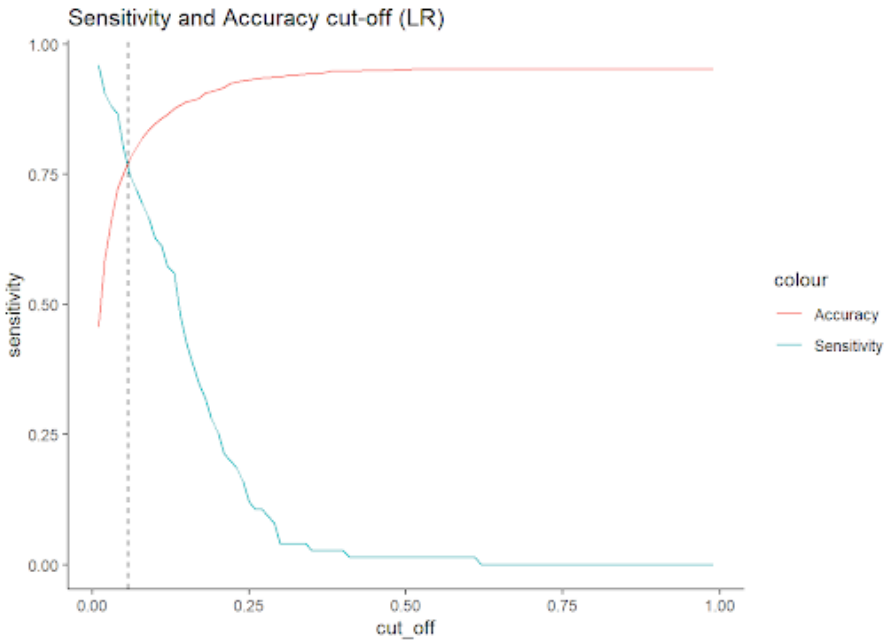
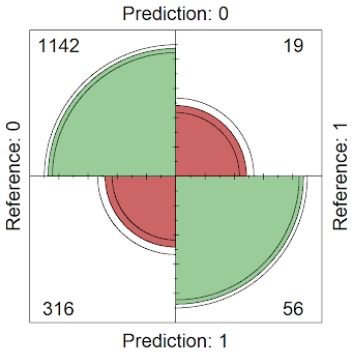


Fig. 4. Table 3.2: LR - Final Confusion matrix



We see an enhancement of our results compared to the initial Confusion Matrix. The accuracy is of 78.15% and the sensitivity is of 74.667%. Both the accuracy and sensitivity tell us that the validation group is doing very well. Around 75% of patients are correctly classified as having a stroke. As for the false negatives they are of $19/1533 = 1.24\%$ and the false positives are of $316/1533 = 20.6\%$. We already see that "only" 19 stroke patients are dangerously classified as non-stroke, which is a relatively small percentage of the total sample, for 316 non-strokes misclassified, which is relatively correct. Our model works rather well with the validation set.

4.2 *Classification Tree*

We start by plotting the full tree and extracting the main root nodes/variables that separate the observations: a full tree is built, and its important variables found (see in Appendix: Graph 2: CT - Importance of variables). Then, we proceed to finding the best cp, the one that maximizes both accuracy and sensitivity, to create our best pruned tree (see in Appendix: Graph 5: CT - Sensitivity and Accuracy). In our case, this returns nearly the same tree as before- the full tree. It seems logical: since our two stroke classes are unbalanced, with no absolute distinctive features, the tree needs more branches in order to precisely separate both classes.

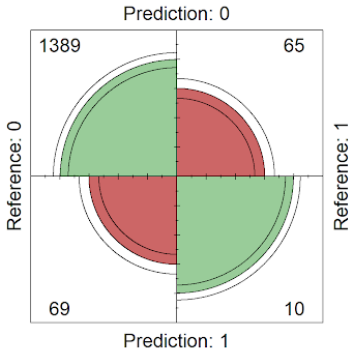


Fig. 5. Graph 7.2: CT - Confusion matrix

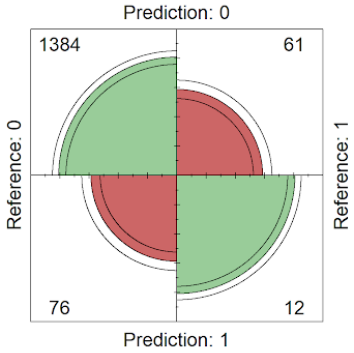
With our best pruned classification tree we have an accuracy of 91.26% and a sensitivity of 0.1333%. Based only on accuracy measure, the CT model has a good predictive power, because it is very good at detecting non-strokes (specificity of 95.25%). However, the goal of the model is to predict stroke occurrences. In that sense, the classification tree performs extremely low, when dealing with actual stroke predictions. As for the false negatives they are of $65/1533 = 4.24\%$ and the false positives are of $69/1533 = 4.5\%$. When trying to impute another classification rule by finding the optimal cut-off to apply to the CT predictions propensities (see in Appendix: Graph 8: CT - Plot Sensitivity and Accuracy cut-off), the CT tree with cut-off (0.01) stays the same, unable to predict enough stroke occurrences.

4.3 *K-Nearest Neighbors*

For the K-nn method, we need to find the best k. We plot our performance measures: accuracy, sensitivity and the mix of both (see in Appendix: Graph 1: KNN - Plot Accuracy, Sensitivity, Both). We find that our best k is 2. This is relatively intuitive as a lower k induces more sensitivity to local characteristics. For unbalanced data sets, taking a big k would result in most of the outcome being attributed to the majority class, in our case all observations would be mislabeled as non-stroke outcomes.

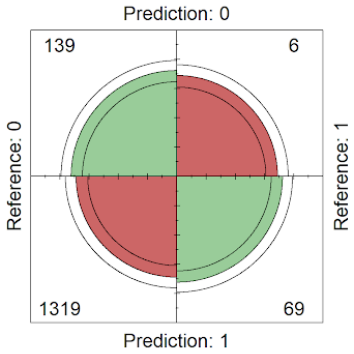
In our confusion matrix for the K-nn method we see that : The accuracy is 91.06% and the sensitivity 0.164%. We come close to the same conclusions than for the classification method. Accuracy is high, because the model classifies most observations as non-strokes, due to the outcome unbalance. As for the false negatives they are 76/1533 = 4.96% and the false positives 61/1533 = 3.98%. By default, K-nn has a high rate of false negatives.

Fig. 6. Graph 2.2: KNN - Initial Confusion matrix



When changing the classification rule applied to K-nn predictions propensities, the best cut-off is at 0.51 (see in Appendix: Graph 3: KNN- Plot Sensitivity and Accuracy cut-off). The new confusion matrix, with a 0.51 cut-off, inverts the situation: it classifies most observations as strokes instead of non-strokes. Its accuracy is 13.57% and its sensitivity is 92% . Sensitivity has mar-

Fig. 7. Graph 4.2: KNN - Final Confusion matrix



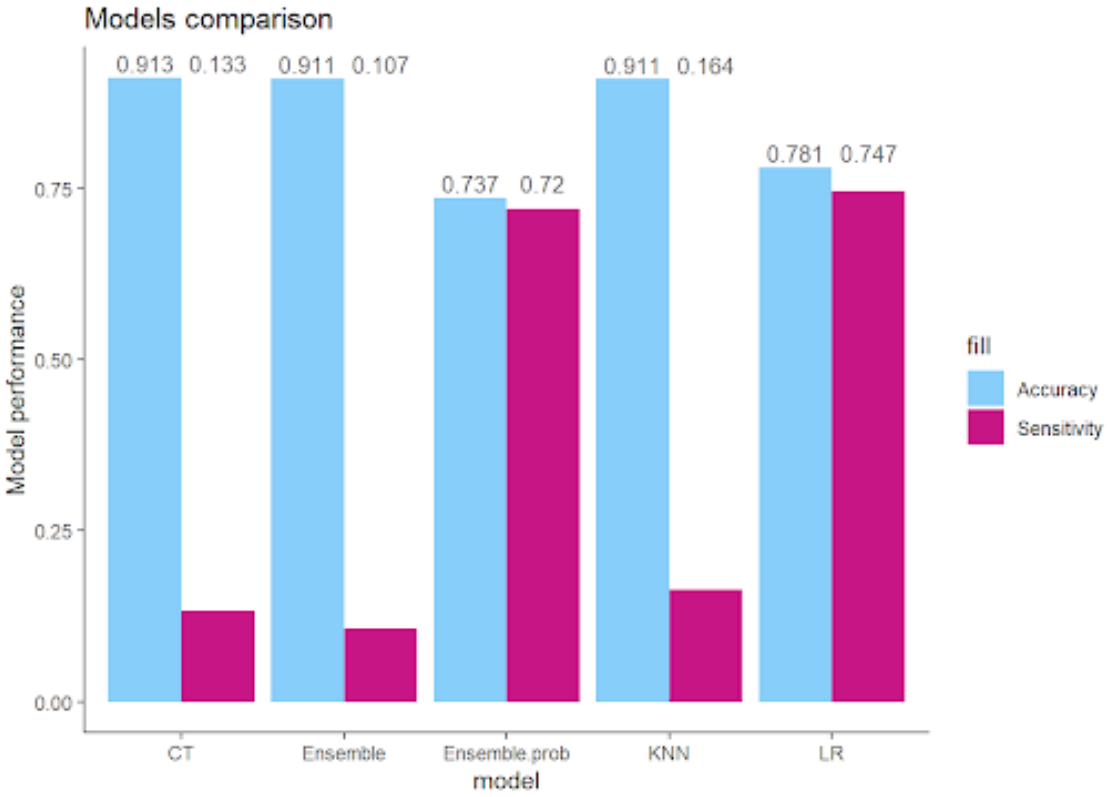
vellously increased, but to a costly price in ac-

curacy: more than 1300 patients would be cared for as if they are stroke liable, when they are not. This results in a financially costly, time-consuming and effort-demanding situation from the hospital system. This is why an equilibrium is important.

4.4 *Ensembles methods and Model comparison*

After finding the best cut-off (Accuracy-Sensitivity mix) to use for the average ensemble, we compute for each observation the majority vote and the average probability of belonging to class stroke, following our cut-off rule. Then, we can compare models: LR (with cut-off), CT (with cut-off), K-nn (with cut-off), and the two ensembles.

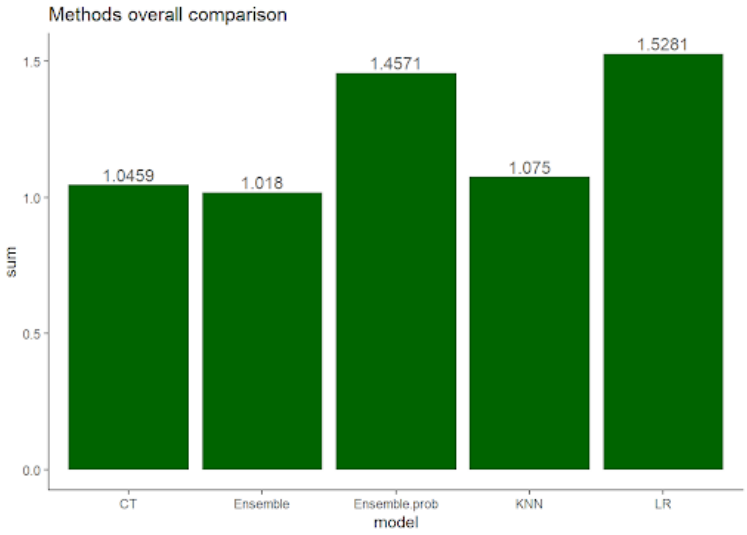
Graph 3.3: Ensembles - All models' accuracy & sensitivity



Our model comparison graph shows all models' accuracy and sensitivity. At first glance, we see that CT, Ensemble (binary) and K-nn models are good as accuracy models, but are low in sensitivity. On the other hand, K-nn, with its 0.51 cut-off, scores high on sensitivity, but is overall inaccurate. Our accuracy and sensitivity measures will determine which model is the most appropriate to build. Thus, we assess models' performance via the sum of both accuracy and sensitivity below.

Since it does not allow for a different classification rule that mirrors the misclassification costs of the classes, the ensemble method has the lowest Accuracy-Sensitivity mix. The CT model and K-nn models represent two extremes: both models misclassify almost all observations as either non-stroke (CT) or stroke (K-nn) occurrences. In comparison, the Logistic Regression model and Average Ensemble method have ideal trade-off between accuracy and sensitivity, since they allowed for tailored cut-off. The average ensemble method has proved that the concept of combining models can indeed help improve models predictive power: it performs reasonably well. Still, with its almost perfect trade-off, it seems that Logistic Regression is the final most efficient method for predicting stroke cases in our validation data set. Therefore, we continue our research based on this model, that we will now compare to its reduced version.

Fig. 8. Graph 3.4: ENS - Methods overall comparison

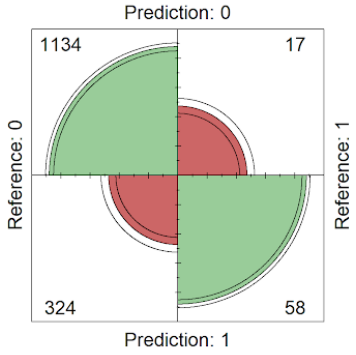


4.5 *Finding the Best Model*

4.5.1 *Model Reduction* Our reduced LR model is built with only the significant variables (i.e age, hypertension, heart_disease and avg_glucose level (at the border of significance)). This is done in order to avoid that our model overfits the data, and in order to reduce the noise due to too many variables being taken into account.

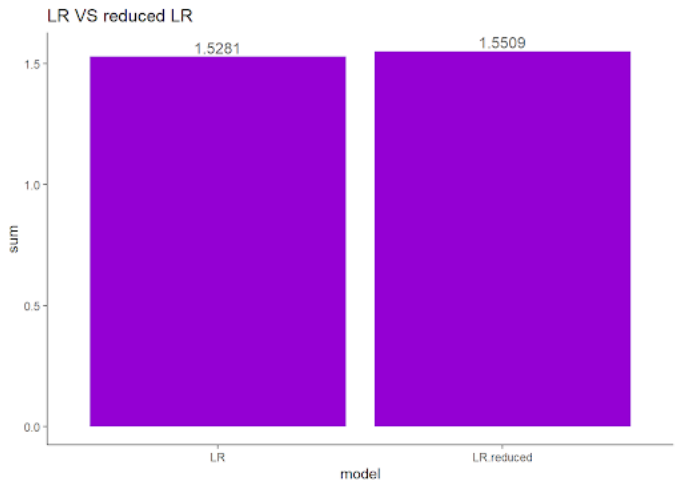
When comparing both models (full vs reduced) on our validation set, we see from its confusion matrix that the reduced model has 2 more stroke predictions (gone from 56 to 58), as well as two less false negatives (gone from 19 to 17). It means our reduced model fits our validation data better. This is to be expected, since the reduced model only focuses on the significant variables that impact the outcome stroke, and discards the "noise" other insignificant variables may bring.

Fig. 9. Table 2.2: Reduced LR - Confusion matrix



While looking at the accuracy-sensitivity mix, we observe that in both cases the reduced logistic regression fits our data better.

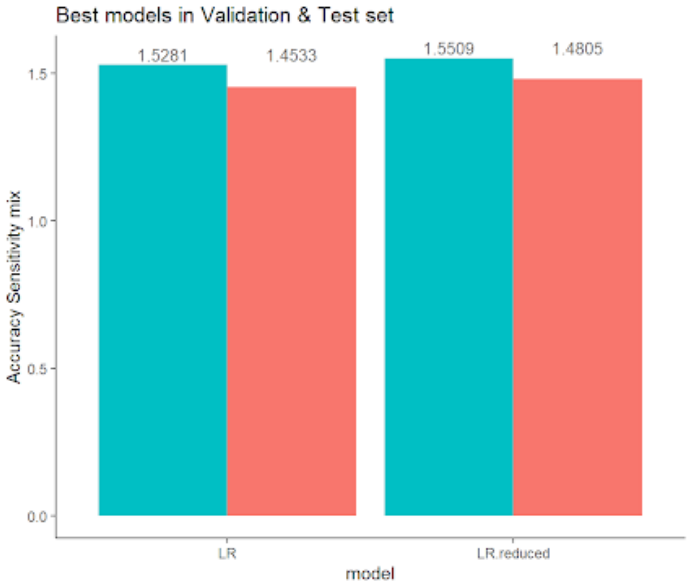
Moreover, the diminished model performs better with the validation set. This is to be expected, since the reduced model is remodeled and redesigned to adjust our validation set, to try and fit it (= classify its observations) in the most correct way. This enhancement is efficient in finding a bet-



ter predictive model.

4.5.2 *Cross Validation* By using the test set, we test the performance of the best built model and can measure its real efficiency on new data, in order to respond to the question: is our best model only good at explaining and fitting the existing data (classification), or can it also make good predictions when new data are injected (profiling) ?

Fig. 10. Graph 3: Test set - Model Comparison (LR + Reduced)



As we observe from the comparison graph, both the default LR and the reduced LR model work a bit better on the validation set, on which they were constructed. Still, they work pretty well on the test set. The reduced model, in both cases, works even better than the default LR. Thus, we can confidently state that out best model, the reduced LR one, has both explanatory and predictive uses: it can classify well enough, as well as profile stroke records.

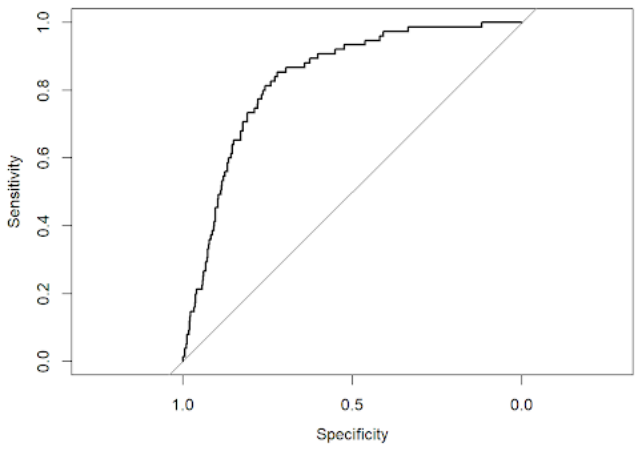
4.6 *Lift, Decile charts and ROC Curve*

To assess the extent of our best model's predictive power, we can use gain chart. A Lift & Decile chart (see in Appendix: Graph 4.1: Best model - Gain & Decile chart) tells us if the model's

predictive power is better than random, by comparing it to a baseline random model. The lift chart measures the efficiency of the model by computing the ratio of the results with and without model. Our lift chart has quite a good predictive power: its curve is far from the diagonal (results without any model), denoting that we have a good predictive model. The decile chart, that has a top decile higher than 3.5, indicates that choosing the top 10% of the patients that give the highest predicted likelihood of strokes, we would help detect 3.5 more stroke cases compared to choosing 10% of the patients at random. Similar conclusions can be confirmed with a ROC curve below.

By using the ROC Curve, we can visualise both sensitivity (true positives) and specificity (true negatives) on the same graph and see their evolution according to a decided threshold. The naive rule is the diagonal line between both the starting and ending point. The more our curve is further to the left, the better our model's predictive power. In our case, we see that the curve is steeper on the left side of the graph, signifying a good performance. We also observe a big area under the ROC curve, indicating that we have a high overall accuracy with our best model (Ltd, 2021).

Fig. 11. Graph 4.2: Best model - ROC curve

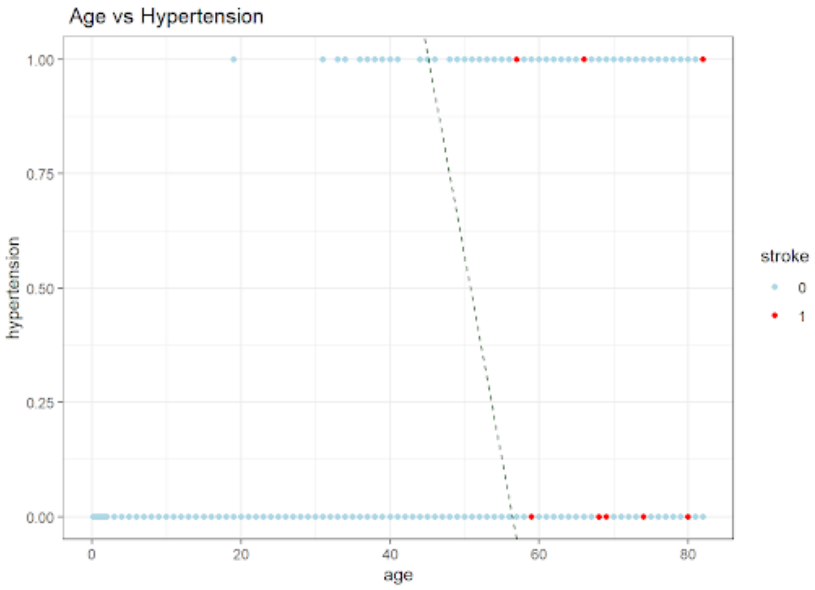


4.7 *Linear Discriminant Analysis*

Lastly, we briefly introduce how Linear Discriminant Analysis (LDA) method could be used in our situation. From the best model (reduced LR), the significant predictors are in order (from

most significant to least): age, hypertension, heart_disease, and avg_glucose_level. Plotting the three most significant variables can be done on a 3D graph with a plan separating the classes. As a case study, we use only two variables, in order to have a 2D graph. We plot the most significant variables, age and hypertension, one against another by class (stroke) on the same graph.

Fig. 12. Graph 2: LDA - Scatterplot with DA line



The higher the age, the higher the occurrences of strokes (more red dots on the right). In case of hypertension, the likelihood of strokes is also higher (a bit more red dots top right). The DA line is at 110°, indicating that the top right observations would classify as stroke, therefore confirming that hypertension being 1, and higher age levels have positive impact on stroke outcome.

Since the cost of misclassifying a stroke is different from the cost of misclassifying a non-stroke, we want to minimize the expected cost of misclassification rather than the simple error rate, which does not account for unequal misclassification costs. We consider:

- q(0) the cost of missclassifying a class 0 member (into class 1);
- q(1) the cost of missclassifying a class 1 member (into class 0);

The cost must be integrated into the constants of the classification functions by adding log(q1) to the constant for class 1 (stroke). Thus, the score for class 1 membership would increase:

observations become more easily classified as stroke (= belonging to class 1) with the modified (increased) constant coefficients, in order to avoid the cost of a mistake (= misclassifying stroke as non-stroke).

To determine the relationship cost requires domain knowledge: is classifying a non-stroke as stroke 20, 30, 40 times higher, than not detecting it? Once it is determined or explored by experts that could accurately estimate it, the LDA model can work to its full extent with a tailored classification rule, and could even represent a good alternative to Logistic Regression models, especially in cases when few variables are proven significant.

5. Conclusion

In conclusion, while classification tree and K-nn algorithms give mixed results, with the logistic regression technique, we achieved greater balance with an Accuracy-Sensitivity mix. The sum of accuracy and sensitivity was used to assess models' performance. This procedure allowed us to build and select the most balanced classification model: one that is able to predict the majority of stroke cases without too many false positives.

When predicting the likelihood of a new patient getting a stroke, our best model, the reduced Logistic Regression, has a 70% chance of correctly classifying the patient as a stroke outcome, without too many mishaps: its general accuracy allows for correct predictions 78% of the time. The assumption that medical terms have higher significance than social terms is pretty much satisfied in our data modeling, that indicated age, heart disease, hypertension and average glucose level as the significant variables on the likelihood to get a stroke .

Our model enabled us to create the profile of a typical patient that suffers from stroke. It corresponds to an older person, aged around 70 in average, that suffers from hypertension and has a heart disease. A higher average glucose level also tends to be an indicator. All other characteristics were found irrelevant by our model, based on the stroke data set we analysed.

6. Recommendations

Following the conclusion of our research paper, we can advise a few practices, via concrete and technical recommendations.

The conclusions of our research paper must ultimately be applied in the real world. At last, we must respond our final research questions: from our data, which strategy/what policies would be the most effective for a public health campaign ? As a government, it is important when putting out a campaign, especially a healthcare one, to target the right people. Indeed, campaign projects require tremendous investments in money, time and efforts. As we have seen throughout our research paper and data set analysis, the typical stroke victim is an older person, that suffers from hypertension, and has a heart disease. Therefore, health authorities should better invest in targeting this demographic, finding the best way to contact and help spread awareness. The first group to target is the impacted patients: those likely to get a stroke. One first and obvious mean of communication would be directly via the profiled patient's doctor. Large platforms could be used, like posters in hospital facilities, inviting people to get their hypertension and heart diseases checked. It is said that older people, that are more prone to health problems, frequent hospitals and/or their doctors more regularly. A typical, more precise, profile of the people directly concerned could be created: if older people tend to read newspapers, the medium could be used to create visuals on, directly communicating the prevention information to the target group. Gender could also be targeted differently if significant differences are observed: if older gentlemen tend to read sports magazines, while older ladies listen to the radio a lot, those mediums should be targeted, using efficient usual marketing techniques. The second group of people closest to the target audience, are those around them (e.g. family, friends). They could be the one advertising healthier practices and saving lives. The third group to target is the general population. Strokes are one of the top causes of death. As a result, it is important to spread awareness to all future generations. In Switzerland, a first aid paper is required when demanding

a driver's license, that 68% of the population possesses (Office, 2018). Incorporating prevention into this mandatory training is one solution. With the democratized popularity of social media, a good way to inform young populations would be to use those platforms in interactive ways.

For the theoretical shortcomings of our research paper, one is the definition of a stroke used. There are multiple types. For a more efficient analysis and understanding, it could be useful to know which one our data set is based on.

As for the technical shortcomings, it is easier to construct an efficient model on balanced outcomes as it simplifies the classification mechanisms used (default cut-offs). One thing that could have been done differently is re-sampling, instead of modifying the classification rule as we did in our research paper. To re-scale our data set according to the outcome can be a way to reduce the gap between stroke and non-stroke occurrences. Re-sampling methods could also be applied to the other variables of interest that have asymmetrical distribution: separating them into comparable sizes to put their significance in perspective. Also, even though only four variables were proven significant by our final model, it does not mean that others should be totally disregarded as explanatory characteristics of a patient getting a stroke. As an example, though being male has no significant effect on the propensity of get a stroke based on our data set, it does not discard it completely from being a risk factor. Cross-validating researches could be undertaken. Ideally we wouldn't have unknown categories in the data set, for smoking_status or NA's for the bmi variable. Redoing a more attentive survey, that records all characteristics more religiously, and tries to find more stroke occurrences to balance records could also be an option. Also, the LDA technique has only been introduced, since it requires more medical domain knowledge to determine the true misclassification cost that actually responds to one core question of our research paper: how many healthy patients do we accept to misclassify, in order to maybe prevent a single one from having a stroke?

7. Appendix

Part 1: Exploratory Data Analysis (EDA)

Table 1.1: EDA - Initial Data Structure

```
## 'data.frame':    5110 obs. of  12 variables:
##  $ id               : int  9046 51676 31112 60182 1665 56669 53882 10434 2741
9 60491 ...
##  $ gender           : chr  "Male" "Female" "Male" "Female" ...
##  $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
##  $ hypertension     : int  0 0 0 0 1 0 1 0 0 0 ...
##  $ heart_disease    : int  1 0 1 0 0 0 1 0 0 0 ...
##  $ ever_married     : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ work_type        : chr  "Private" "Self-employed" "Private" "Private" ...
##  $ Residence_type   : chr  "Urban" "Rural" "Rural" "Urban" ...
##  $ avg_glucose_level: num  229 202 106 171 174 ...
##  $ bmi              : chr  "36.6" "N/A" "32.5" "34.4" ...
##  $ smoking_status   : chr  "formerly smoked" "never smoked" "never smoked" "s
mokes" ...
##  $ stroke           : int  1 1 1 1 1 1 1 1 1 1 ...
```

Table 1.2: EDA - Standardized Data Structure

```
## 'data.frame':    5110 obs. of  11 variables:
##  $ gender           : Factor w/ 3 levels "Female","Male",..: 2 1 2 1 1 2 2 1
1 1 ...
##  $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
##  $ hypertension     : int  0 0 0 0 1 0 1 0 0 0 ...
##  $ heart_disease    : int  1 0 1 0 0 0 1 0 0 0 ...
##  $ ever_married     : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 1 2 2 ...
##  $ work_type        : Factor w/ 5 levels "children","Govt_job",..: 4 5 4 4 5
4 4 4 4 4 ...
##  $ Residence_type   : Factor w/ 2 levels "Rural","Urban": 2 1 1 2 1 2 1 2 1 2
...
##  $ avg_glucose_level: num  229 202 106 171 174 ...
##  $ bmi              : num  36.6 NA 32.5 34.4 24 29 27.4 22.8 NA 24.2 ...
##  $ smoking_status   : Factor w/ 4 levels "formerly smoked",..: 1 2 2 3 2 1 2
2 4 4 ...
##  $ stroke           : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

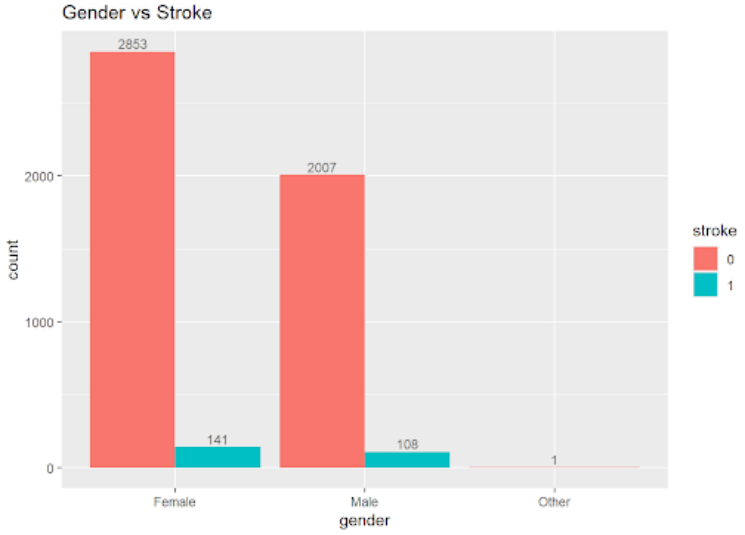
Table 1.3: EDA - Number of different levels for each variable

```
##           gender              age     hypertension     heart_disease
##                3              104                2                2
##     ever_married        work_type   Residence_type avg_glucose_level
##                2                5                2             3979
##              bmi   smoking_status           stroke
##              419                4                2
```

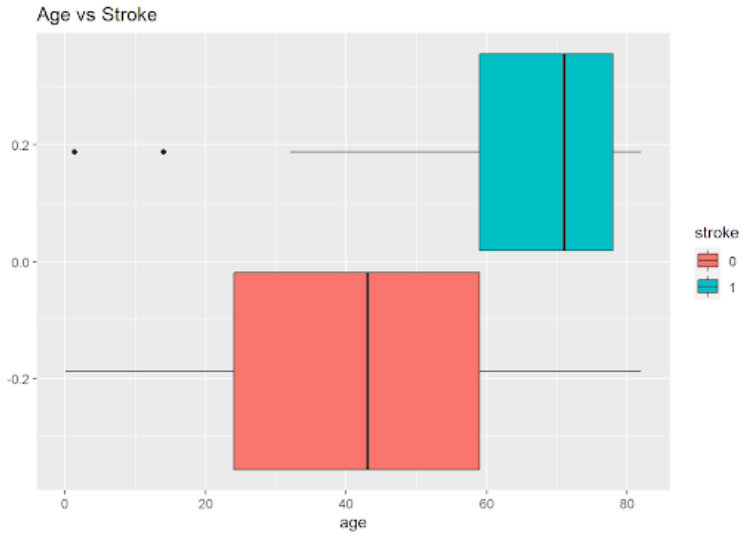
Table 1.4: EDA - Variables summaries

```
##    gender          age         hypertension     heart_disease     ever_marri
ed
##  Female:2994   Min.   : 0.08   Min.   :0.00000   Min.   :0.00000   No :1757
##  Male  :2115   1st Qu.:25.00   1st Qu.:0.00000   1st Qu.:0.00000   Yes:3353
##  Other :   1   Median :45.00   Median :0.00000   Median :0.00000
##                Mean   :43.23   Mean   :0.09746   Mean   :0.05401
##                3rd Qu.:61.00   3rd Qu.:0.00000   3rd Qu.:0.00000
##                Max.   :82.00   Max.   :1.00000   Max.   :1.00000
##
##         work_type    Residence_type avg_glucose_level      bmi
##  children     : 687   Rural:2514     Min.   : 55.12    Min.   :10.30
##  Govt_job     : 657   Urban:2596     1st Qu.: 77.25    1st Qu.:23.50
##  Never_worked :  22                  Median : 91.89    Median :28.10
##  Private      :2925                  Mean   :106.15    Mean   :28.89
##  Self-employed: 819                  3rd Qu.:114.09    3rd Qu.:33.10
##                                      Max.   :271.74    Max.   :97.60
##                                                        NA's   :201
##         smoking_status stroke
##  formerly smoked: 885   0:4861
##  never smoked   :1892   1: 249
##  smokes         : 789
##  Unknown        :1544
##
##
##
```

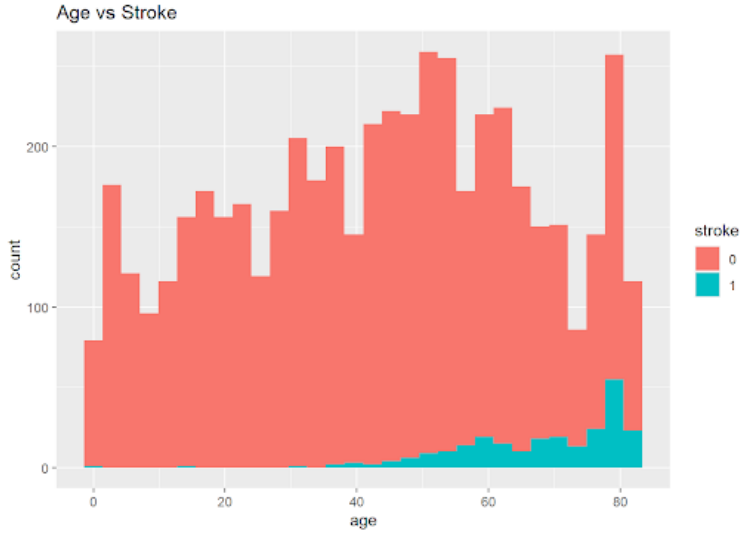
Graph 1: EDA - Data Visualisation: Gender



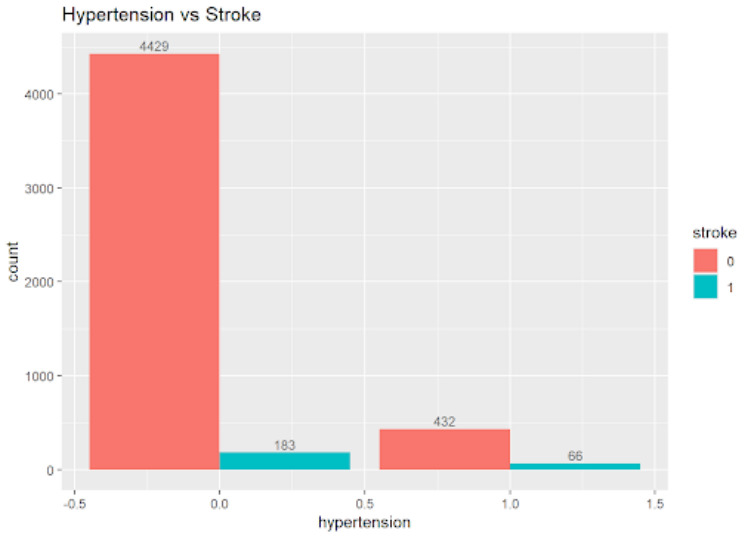
Graph 2.1: EDA - Data Visualisation: Age Box plot



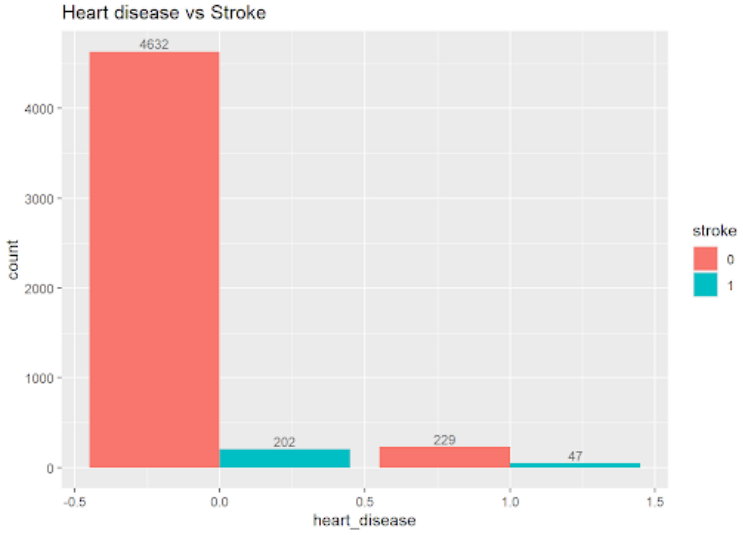
Graph 2.2: EDA - Data Visualisation: Age Histograms



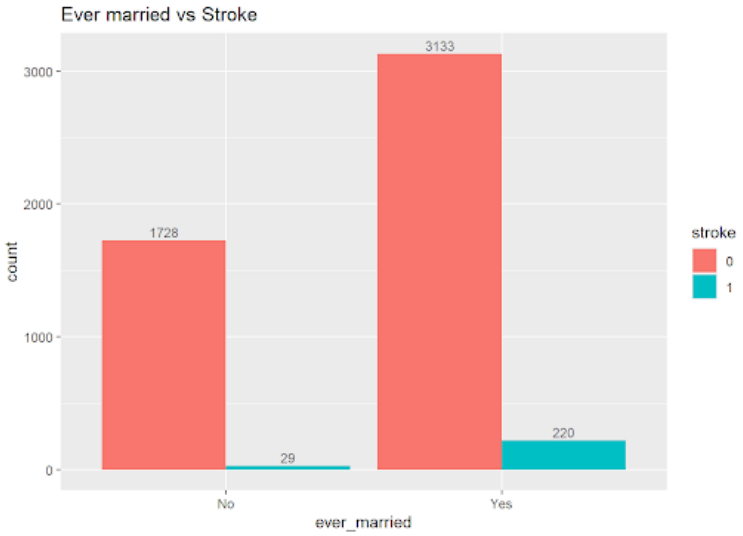
Graph 3: EDA - Data Visualisation: Hypertension



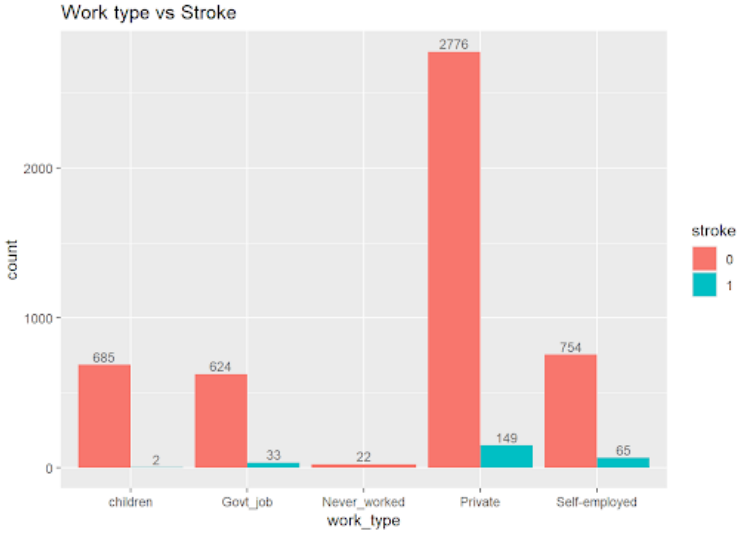
Graph 4: EDA - Data Visualisation: Heart Disease



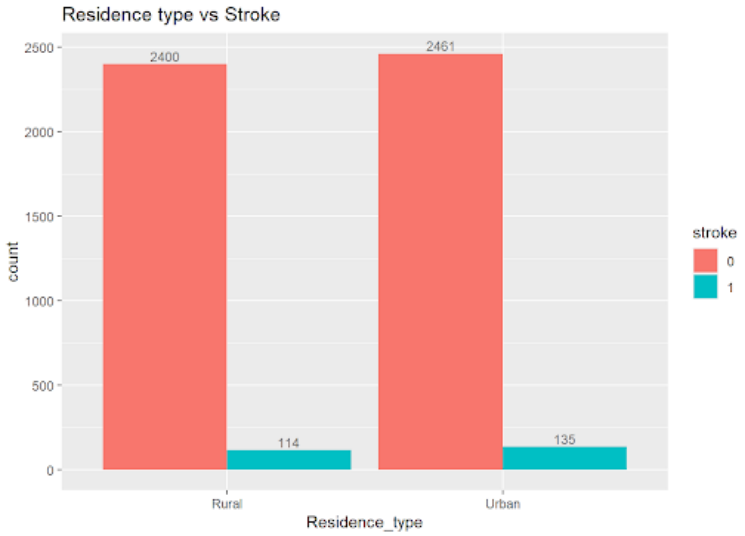
Graph 5: EDA - Data Visualisation: Ever Married



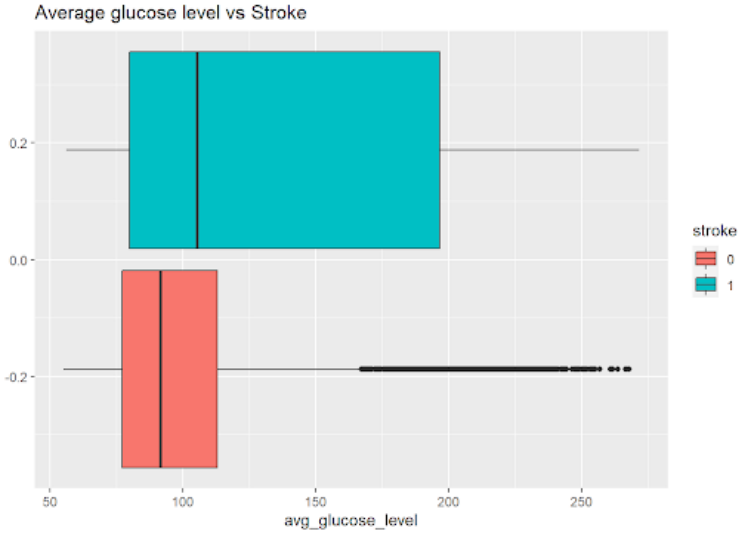
Graph 6: EDA - Data Visualisation: Work Type



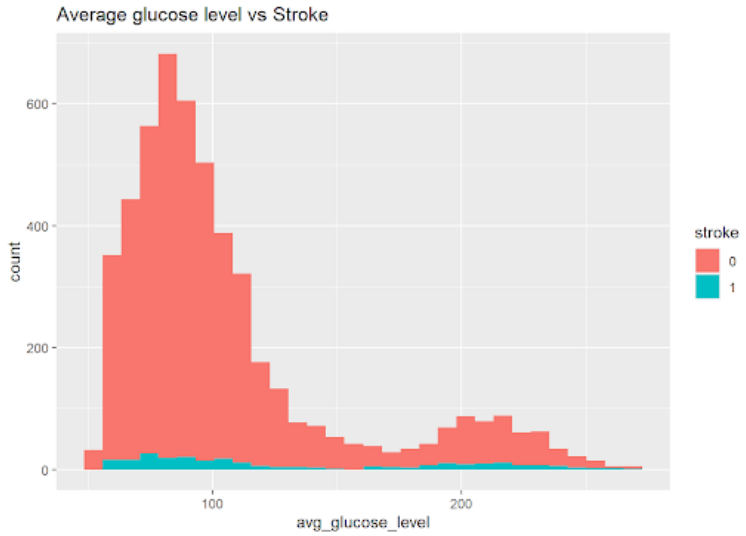
Graph 7: EDA - Data Visualisation: Residence Type



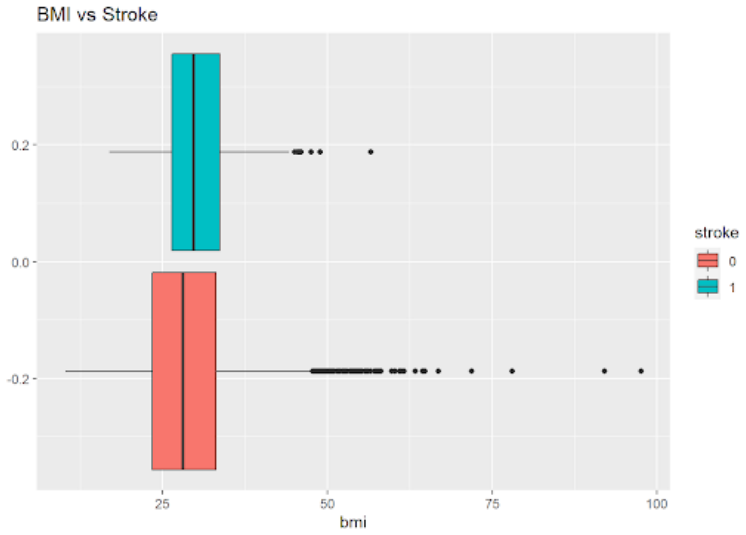
Graph 8.1: EDA - Data Visualisation: Average Glucose Boxplot



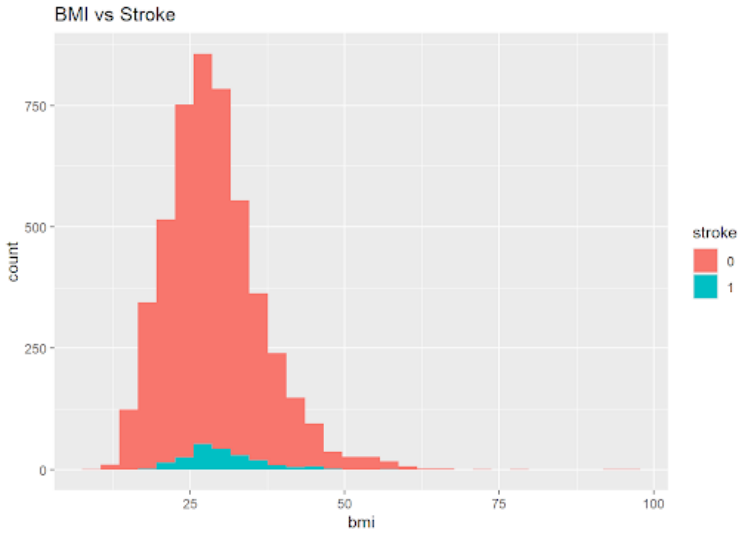
Graph 8.2: EDA - Data Visualisation: Average Glucose Histograms



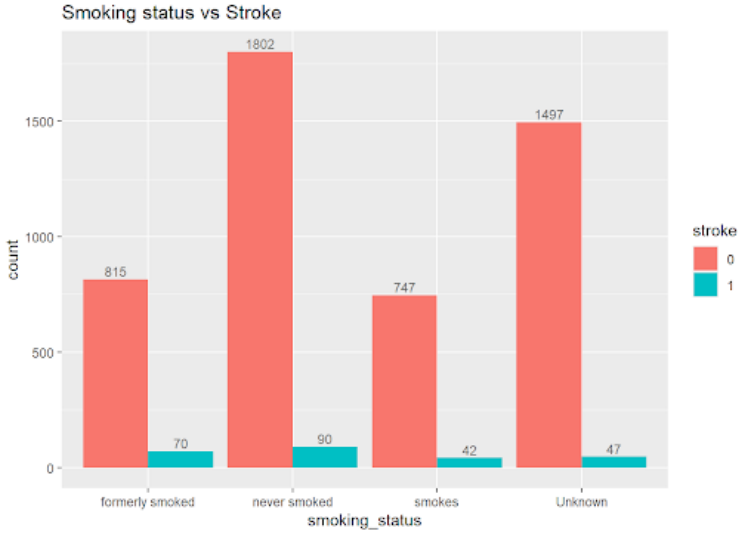
Graph 9.1: EDA - Data Visualisation: BMI Box plot



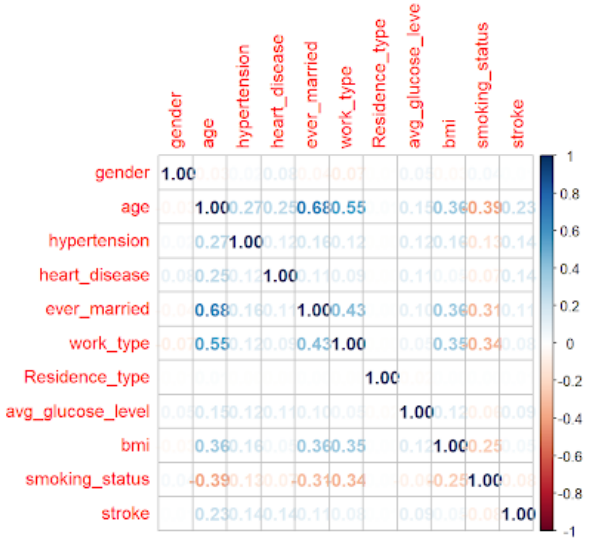
Graph 9.2: EDA - Data Visualisation: BMI Histograms



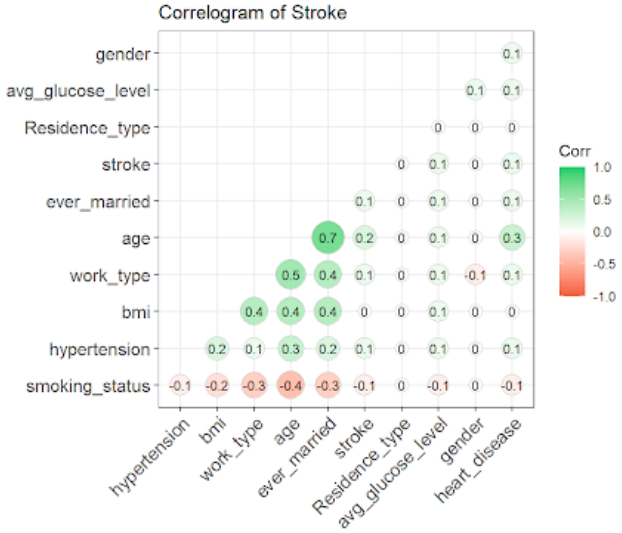
Graph 10: EDA - Data Visualisation: Smoking Status



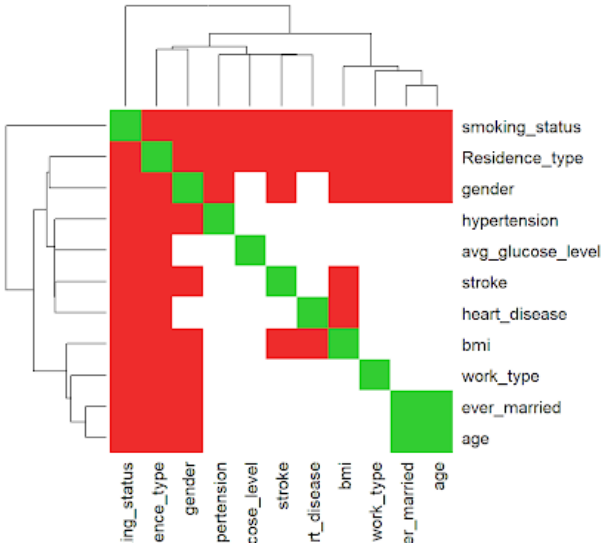
Graph 11.1: EDA - Correlation Matrix 1



Graph 11.2: EDA - Correlation Matrix 2



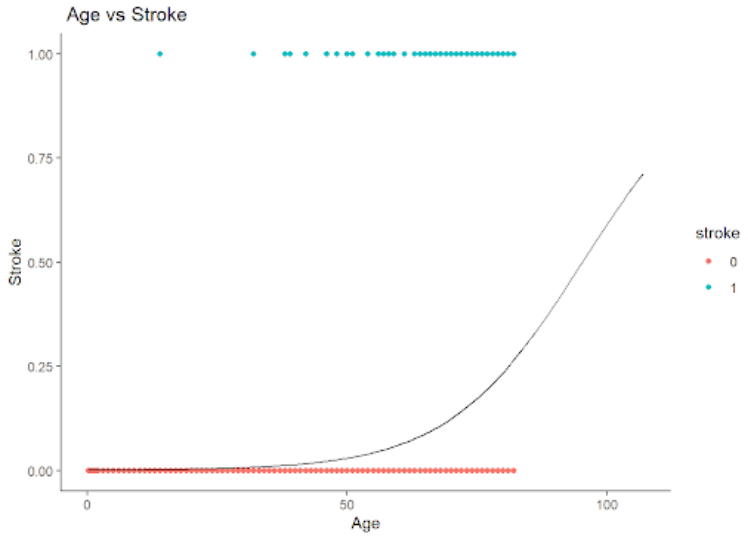
Graph 11.3: EDA - Correlation Matrix: heat map



Part 2: Logistic Regression (LR)

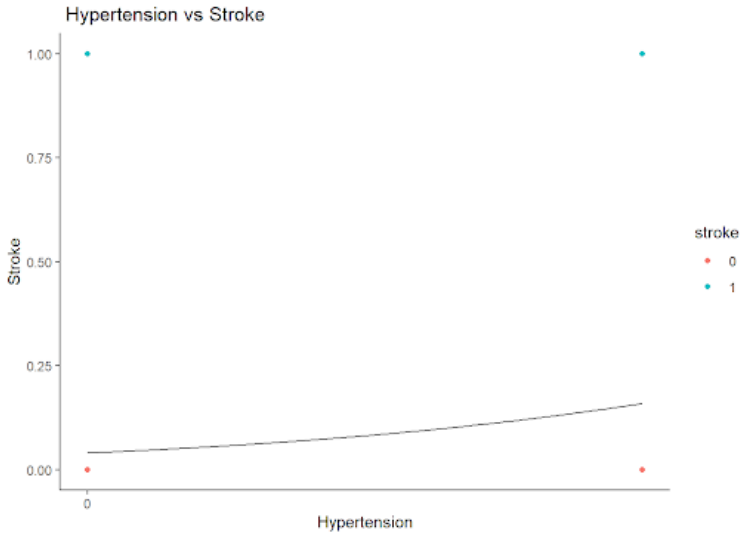
Table 1: LR - Generalized Linear Model

```
##
## Call:
## glm(formula = stroke ~ ., family = "binomial", data = train.lr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1918  -0.3189  -0.1624  -0.0824   3.5431
##
## Coefficients:
##                               Estimate Std. Error z value            Pr(>|z|)
## (Intercept)                  -6.871647   1.109622  -6.193      0.000000000591
## genderMale                   -0.079608   0.198349  -0.401             0.68816
## age                           0.075445   0.008411   8.970 < 0.000000000000002
## hypertension                  0.714353   0.224014   3.189             0.00143
## heart_disease                 0.573700   0.251742   2.279             0.02267
## work_typeGovt_job            -0.891500   1.165161  -0.765             0.44419
## work_typeNever_worked       -11.217690 638.089433  -0.018             0.98597
## work_typePrivate             -0.887875   1.143017  -0.777             0.43729
## work_typeSelf-employed       -1.263710   1.178934  -1.072             0.28376
## Residence_typeUrban           0.165124   0.193104   0.855             0.39249
## avg_glucose_level             0.003068   0.001704   1.801             0.07174
## bmi                          -0.003033   0.015891  -0.191             0.84865
## smoking_statusnever smoked   -0.112022   0.248067  -0.452             0.65157
## smoking_statussmokes          0.139040   0.314279   0.442             0.65819
## smoking_statusUnknown         0.203656   0.284873   0.715             0.47467
##
## (Intercept)                 ***
## genderMale
## age                         ***
## hypertension                **
## heart_disease               *
## work_typeGovt_job
## work_typeNever_worked
## work_typePrivate
## work_typeSelf-employed
## Residence_typeUrban
## avg_glucose_level           .
## bmi
## smoking_statusnever smoked
## smoking_statussmokes
## smoking_statusUnknown
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1033.36  on 2553  degrees of freedom
## Residual deviance:  803.86  on 2539  degrees of freedom
## AIC: 833.86
##
## Number of Fisher Scoring iterations: 15
```

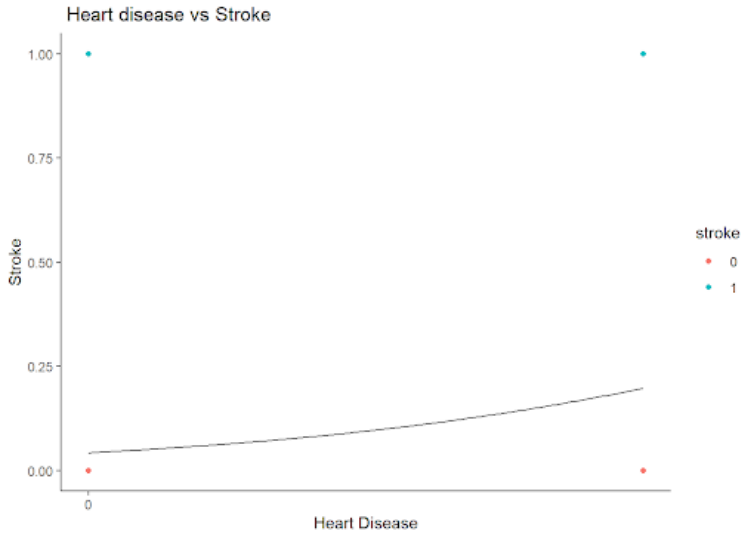


Graph 1: LR - Age vs Stroke

Graph 2: LR - Hyper tension vs Stroke



Graph 3: LR - Heart Disease vs Stroke



Graph 4: LR - Average Glucose Level vs Stroke

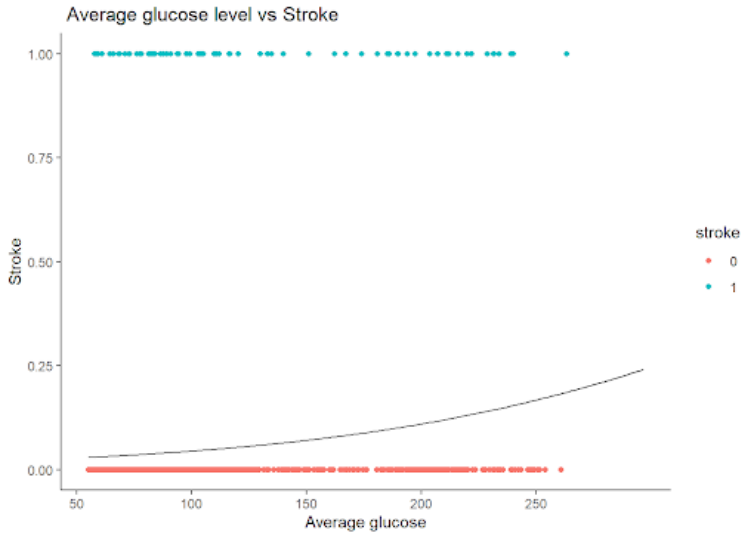
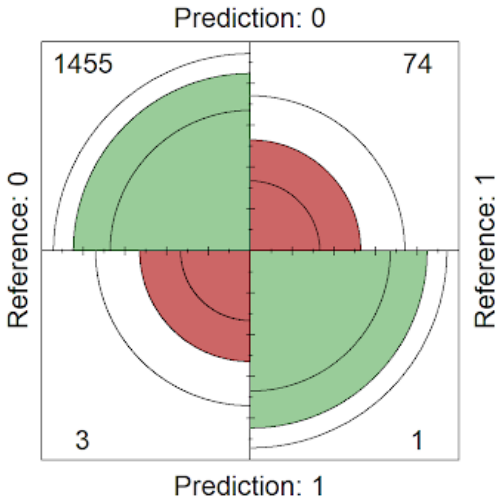


Table 2.1: LR - Initial confusion matrix

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1455   74
##          1    3    1
##
##                Accuracy : 0.9498
##                  95% CI : (0.9376, 0.9602)
##     No Information Rate : 0.9511
##     P-Value [Acc > NIR] : 0.6226
##
##                   Kappa : 0.0205
##
##  Mcnemar's Test P-Value : 0.00000000000000001496
##
##             Sensitivity : 0.0133333
##             Specificity : 0.9979424
##          Pos Pred Value : 0.2500000
##          Neg Pred Value : 0.9516024
##              Prevalence : 0.0489237
##          Detection Rate : 0.0006523
##    Detection Prevalence : 0.0026093
##       Balanced Accuracy : 0.5056379
##
##       'Positive' Class : 1
##
```

Table 2.2: LR - Initial Four Fold plot



Graph 5: LR - Plot Sensitivity and Accuracy cut-off

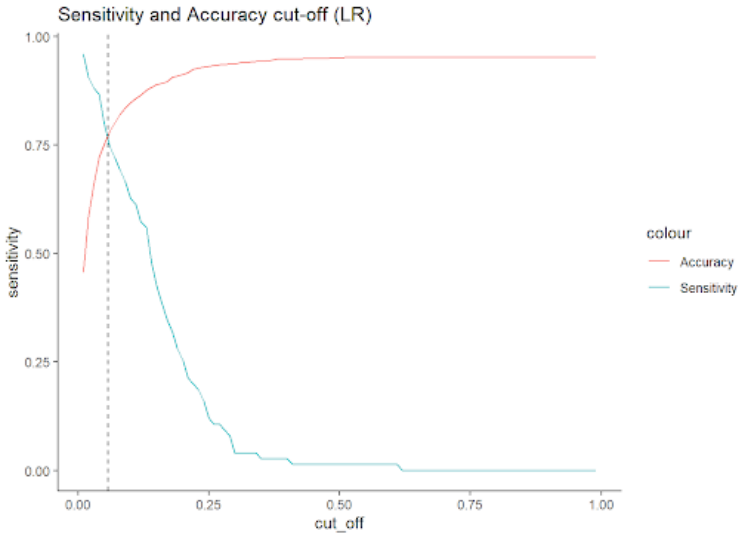
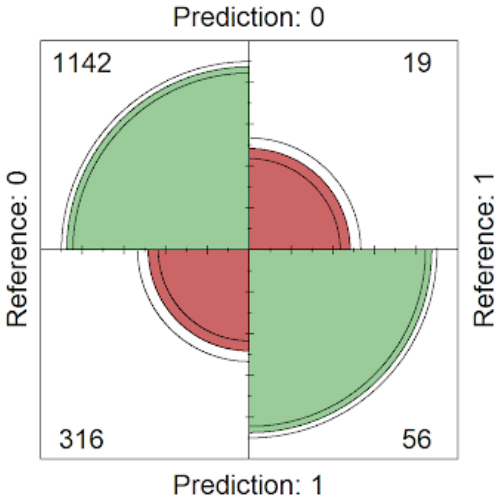


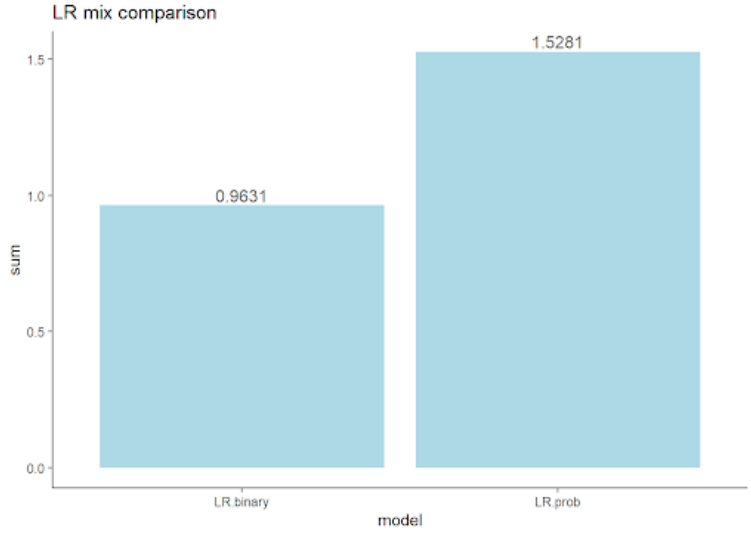
Table 3.1: LR - Final confusion matrix with cut-off

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1142   19
##          1  316   56
##
##               Accuracy : 0.7815
##                 95% CI : (0.7599, 0.8019)
##    No Information Rate : 0.9511
##    P-Value [Acc > NIR] : 1
##
##                  Kappa : 0.1841
##
##  Mcnemar's Test P-Value : <0.0000000000000002
##
##             Sensitivity : 0.74667
##             Specificity : 0.78326
##          Pos Pred Value : 0.15054
##          Neg Pred Value : 0.98363
##              Prevalence : 0.04892
##          Detection Rate : 0.03653
##    Detection Prevalence : 0.24266
##       Balanced Accuracy : 0.76497
##
##        'Positive' Class : 1
##
```

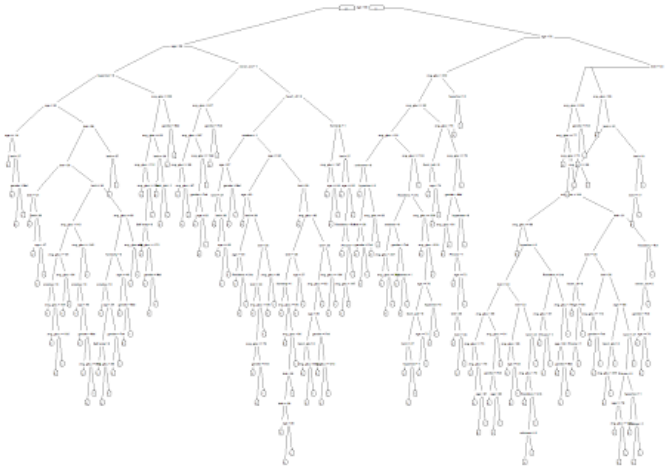
Table 3.2: LR - Final Four Fold plot with cut-off



Graph 6: LR - LR Mix comparison



Part 3: Classification Tree (CT)



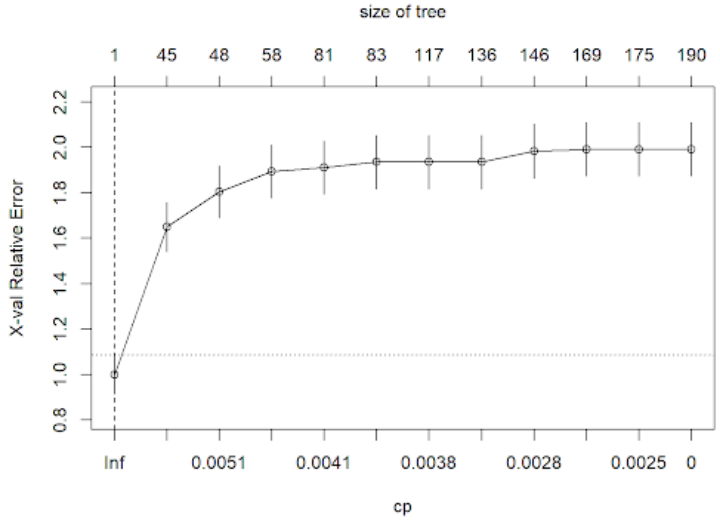
Graph 1: CT - Full tree

Graph 2: CT - Importance of variables

```
## avg_glucose_level               age               bmi            gender
##         86.407429         61.162516         57.178997         20.349436
##           Private      hypertension    Residence_type     Self-employed
##         13.923963         13.647692         11.560232          7.947886
##          Govt_job     heart_disease   formerly_smoked            smokes
##          7.440944          6.714577          6.174170          5.864389
##      never_smoked           unknown
##          4.544718          3.054517
```

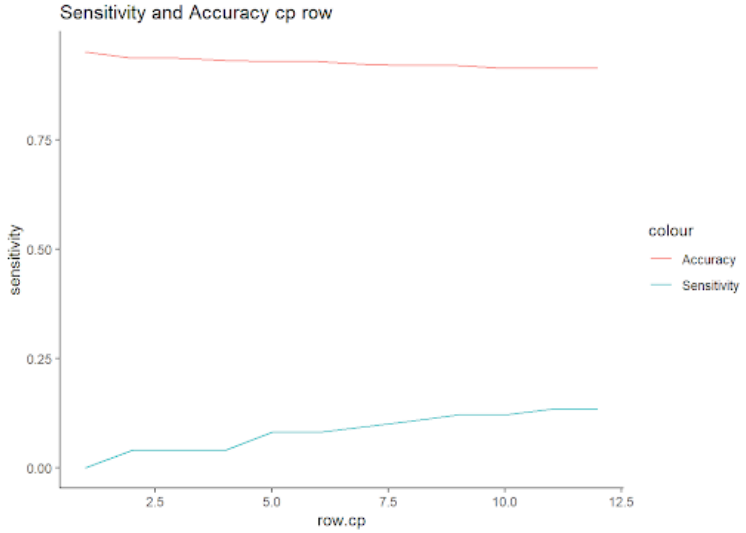
Graph 3: CT - R part function

```
##
## Classification tree:
## rpart(formula = stroke ~ ., data = train.ct, method = "class",
##     parms = list(prior = c(ratio_0, ratio_1)), cp = 0, minbucket = 1,
##     minsplit = 1, xval = 5)
##
## Variables actually used in tree construction:
##  [1] age               avg_glucose_level bmi               formerly_smoked
##  [5] gender            Govt_job          heart_disease     hypertension
##  [9] never_smoked      Private           Residence_type    Self-employed
## [13] smokes            unknown
##
## Root node error: 124.45/2554 = 0.048728
##
## n= 2554
##
##           CP nsplit rel error xerror    xstd
## 1  0.0064826      0  1.000000 1.0000 0.08510
## 2  0.0050891     44  0.618321 1.6492 0.10879
## 3  0.0050891     47  0.603053 1.8036 0.11349
## 4  0.0042414     57  0.526718 1.8939 0.11614
## 5  0.0040285     80  0.389736 1.9104 0.11662
## 6  0.0038168     82  0.381679 1.9346 0.11730
## 7  0.0038168    116  0.251908 1.9346 0.11730
## 8  0.0030534    135  0.167939 1.9346 0.11730
## 9  0.0025445    145  0.137405 1.9834 0.11866
## 10 0.0025445    168  0.061069 1.9914 0.11888
## 11 0.0025445    174  0.038168 1.9914 0.11888
## 12 0.0000000    189  0.000000 1.9914 0.11888
```

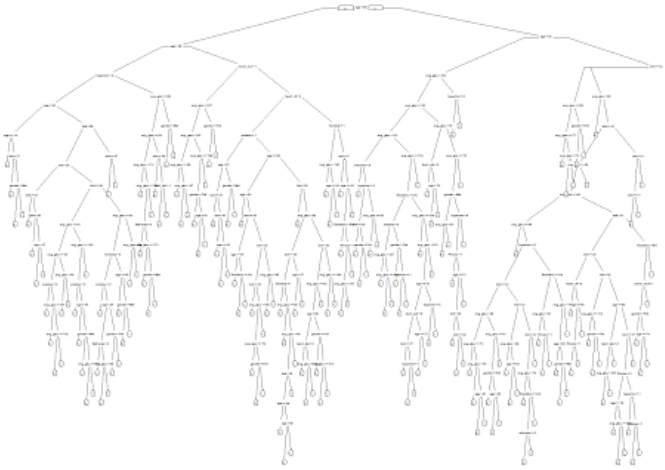


Graph 4: CT - Best CP

Graph 5: CT - Sensitivity and Accuracy



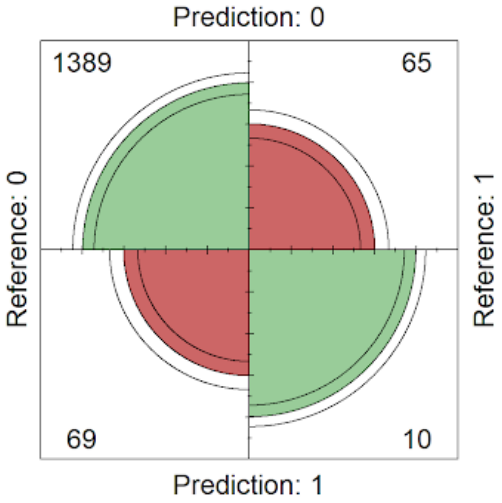
Graph 6: CT - best pruned tree



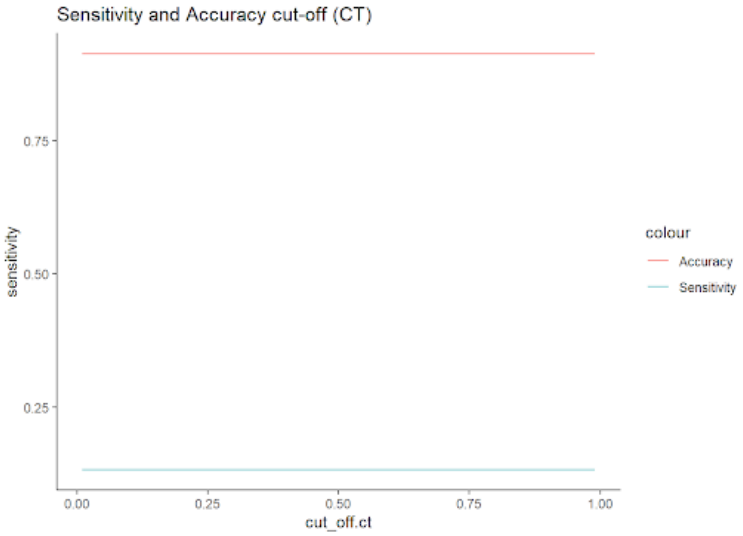
Graph 7.1: CT - Initial confusion matrix

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction    0    1
##          0 1389   65
##          1   69   10
##
##                Accuracy : 0.9126
##                  95% CI : (0.8973, 0.9263)
##     No Information Rate : 0.9511
##     P-Value [Acc > NIR] : 1.0000
##
##                   Kappa : 0.0839
##
##  Mcnemar's Test P-Value : 0.7955
##
##             Sensitivity : 0.133333
##             Specificity : 0.952675
##          Pos Pred Value : 0.126582
##          Neg Pred Value : 0.955296
##              Prevalence : 0.048924
##          Detection Rate : 0.006523
##    Detection Prevalence : 0.051533
##       Balanced Accuracy : 0.543004
##
##        'Positive' Class : 1
##
```

Graph 7.2: CT - Initial Four fold plot



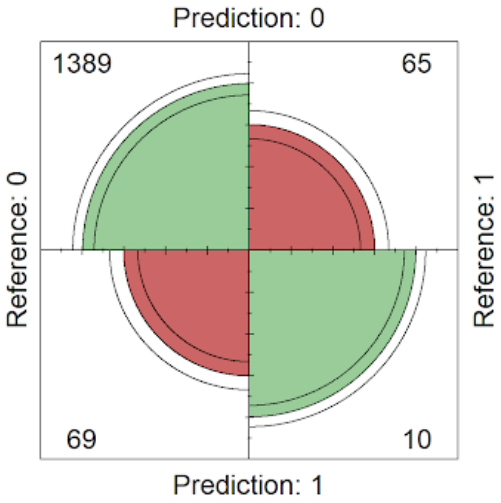
Graph 8: CT - Plot Sensitivity and Accuracy cut-off



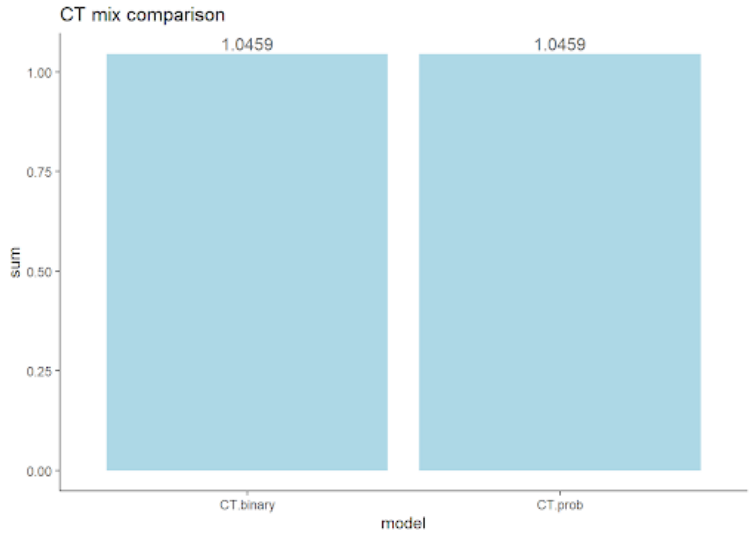
Graph 9.1: CT - Final confusion matrix

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1389   65
##          1   69   10
##
##                Accuracy : 0.9126
##                  95% CI : (0.8973, 0.9263)
##     No Information Rate : 0.9511
##     P-Value [Acc > NIR] : 1.0000
##
##                   Kappa : 0.0839
##
##  Mcnemar's Test P-Value : 0.7955
##
##             Sensitivity : 0.133333
##             Specificity : 0.952675
##          Pos Pred Value : 0.126582
##          Neg Pred Value : 0.955296
##              Prevalence : 0.048924
##          Detection Rate : 0.006523
##    Detection Prevalence : 0.051533
##       Balanced Accuracy : 0.543004
##
##        'Positive' Class : 1
##
```

Graph 9.2: CT - Final Four Fold plot

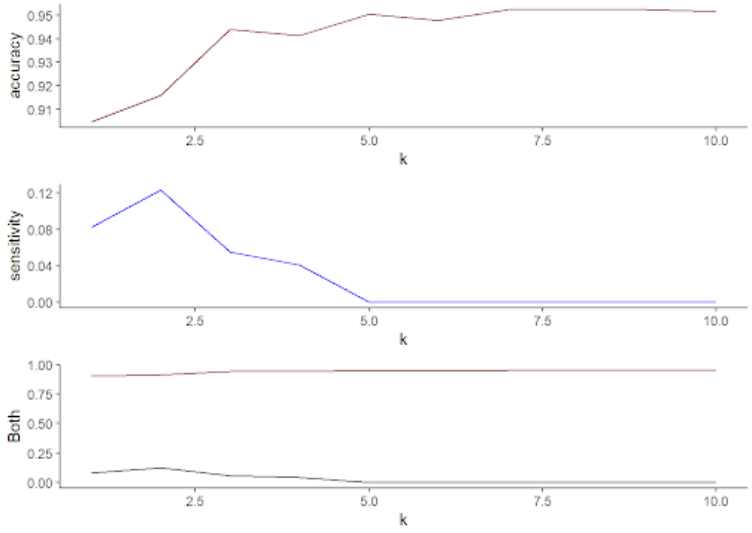


Graph 10: CT - CT mix comparison



Part 4: K-Nearest Neighbors (K-NN)

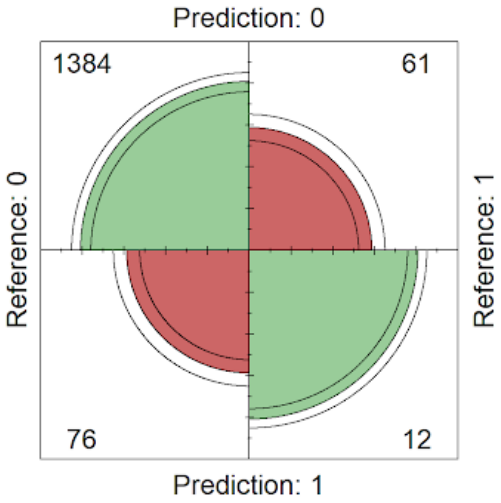
Graph 1: KNN - Plot Sensitivity, Accuracy and Both



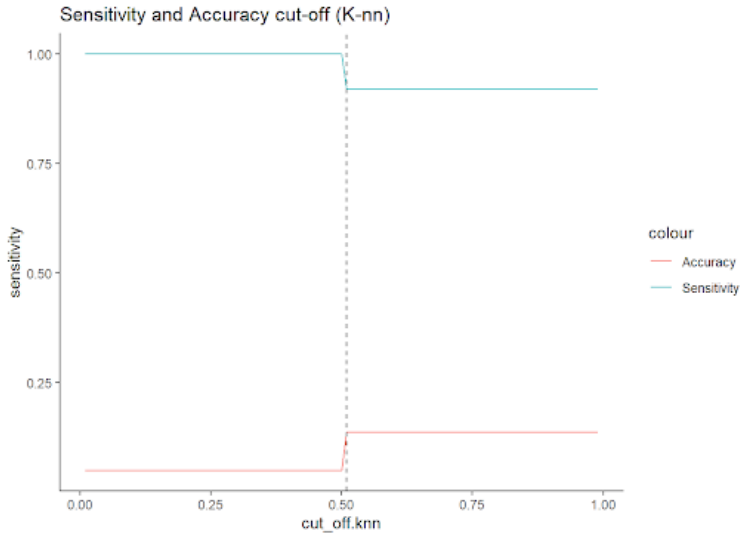
Graph 2.1: KNN - Initial confusion matrix

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1384   61
##          1   76   12
##
##                Accuracy : 0.9106
##                  95% CI : (0.8952, 0.9244)
##     No Information Rate : 0.9524
##     P-Value [Acc > NIR] : 1.0000
##
##                   Kappa : 0.1023
##
##  Mcnemar's Test P-Value : 0.2317
##
##             Sensitivity : 0.164384
##             Specificity : 0.947945
##          Pos Pred Value : 0.136364
##          Neg Pred Value : 0.957785
##              Prevalence : 0.047619
##          Detection Rate : 0.007828
##    Detection Prevalence : 0.057404
##       Balanced Accuracy : 0.556164
##
##        'Positive' Class : 1
##
```

Graph 2.2: KNN - Initial Four Fold plot



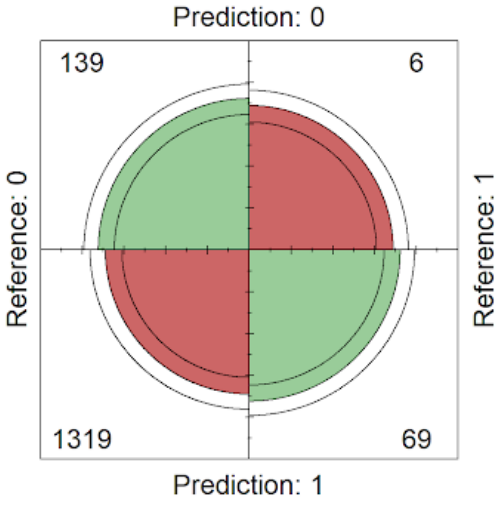
Graph 3: KNN - Plot Sensitivity and Accuracy cut-off



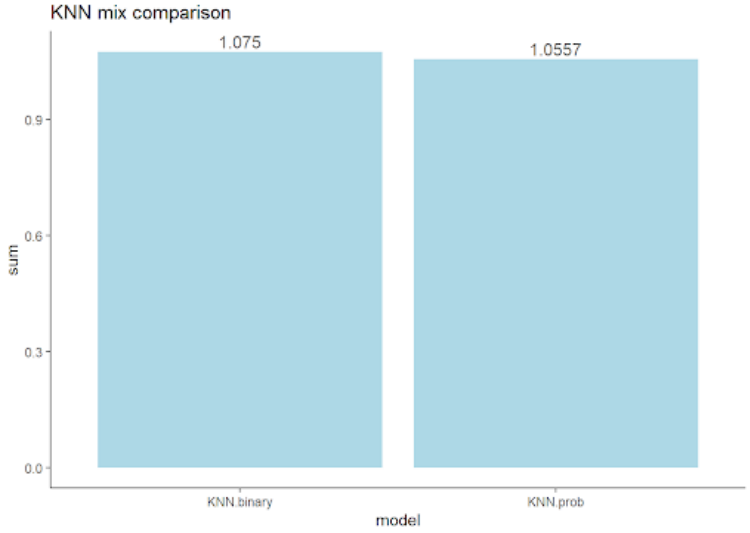
Graph 4.1: KNN - Final confusion matrix

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction    0    1
##          0  139    6
##          1 1319   69
##
##               Accuracy : 0.1357
##                 95% CI : (0.1189, 0.1538)
##    No Information Rate : 0.9511
##    P-Value [Acc > NIR] : 1
##
##                  Kappa : 0.0016
##
##  Mcnemar's Test P-Value : <0.0000000000000002
##
##            Sensitivity : 0.92000
##            Specificity : 0.09534
##         Pos Pred Value : 0.04971
##         Neg Pred Value : 0.95862
##             Prevalence : 0.04892
##         Detection Rate : 0.04501
##   Detection Prevalence : 0.90541
##      Balanced Accuracy : 0.50767
##
##       'Positive' Class : 1
##
```

Graph 4.2: KNN - Final Four Fold plot



Graph 5: KNN - KNN mix comparison

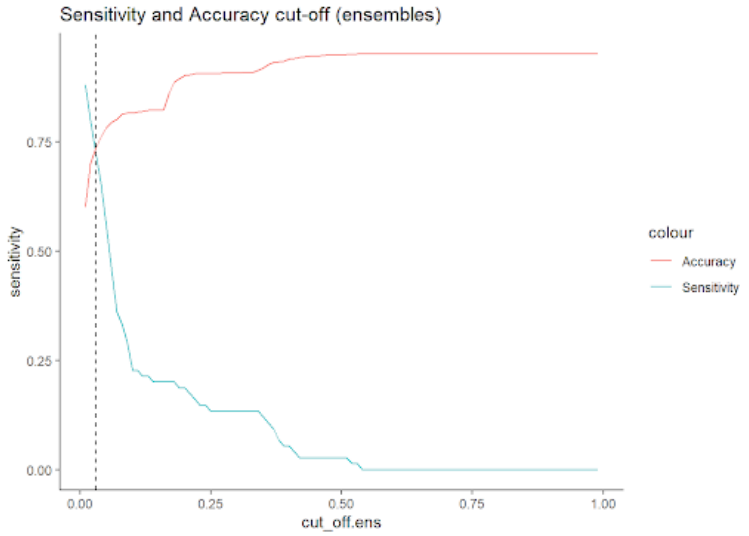


Part 5: Ensemble methods

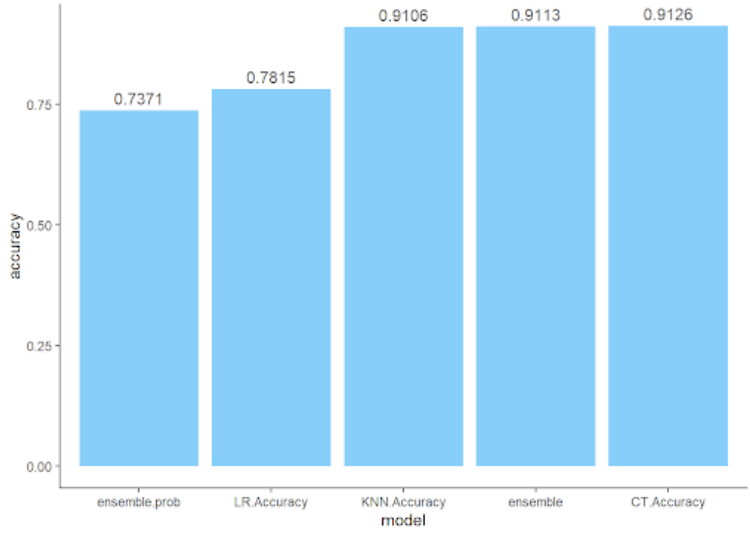
Graph 1: ENS - Head function with first outcome

```
##      actual lr.binary     lr.prob ct.binary ct.prob knn.binary knn.prob
## 4119      0         0 0.011839232         0       0          0        0
## 1380      0         0 0.011537197         0       0          0        0
## 1050      0         0 0.005209831         0       0          0        0
## 1708      0         0 0.004610370         0       0          0        0
## 1087      0         0 0.027404257         0       0          0        0
## 2566      0         0 0.037706301         0       0          0        0
##      average.prob major.voting
## 4119  0.003946411            0
## 1380  0.003845732            0
## 1050  0.001736610            0
## 1708  0.001536790            0
## 1087  0.009161419            0
## 2566  0.012595434            0
```

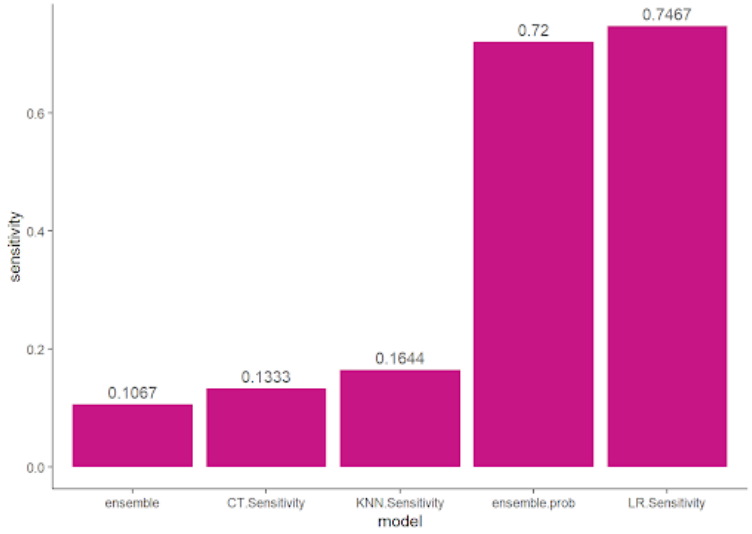
Graph 2: ENS - Plot Sensitivity and Accuracy cut-off



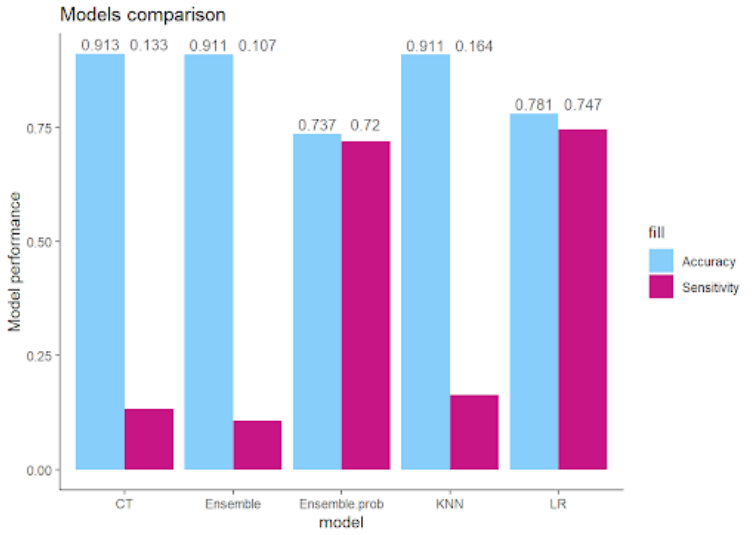
Graph 3.1: ENS - All models' accuracy



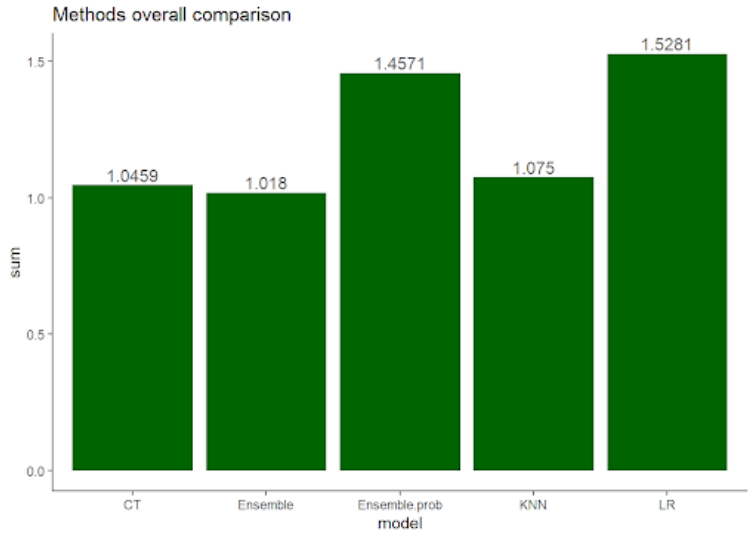
Graph 3.2: ENS - All models' sensitivity



Graph 3.3: ENS - All models' accuracy & sensitivity



Graph 3.4: ENS - Methods overall comparison



Part 6 - Best Model and Cross-Validation (Test set)

Table 1: BEST MODEL - Reduced Generalized Linear Model

```
##
## Call:
## glm(formula = stroke ~ age + hypertension + heart_disease + avg_glucose_level,
##     family = "binomial", data = train.lr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2155  -0.3263  -0.1693  -0.0782   3.7872
##
## Coefficients:
##                   Estimate Std. Error z value             Pr(>|z|)
## (Intercept)      -7.475759   0.512806 -14.578 < 0.0000000000000002 ***
## age               0.069560   0.007352   9.462 < 0.0000000000000002 ***
## hypertension      0.656781   0.218815   3.002              0.00269 **
## heart_disease     0.604553   0.245133   2.466              0.01365 *
## avg_glucose_level 0.003033   0.001654   1.834              0.06663 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1033.36  on 2553  degrees of freedom
## Residual deviance:  809.96  on 2549  degrees of freedom
## AIC: 819.96
##
## Number of Fisher Scoring iterations: 7
```

Table 2.1: Reduced LR - Initial confusion matrix

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction    0    1
##          0 1134   17
##          1  324   58
##
##                Accuracy : 0.7776
##                  95% CI : (0.7559, 0.7982)
##     No Information Rate : 0.9511
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1874
##
##  Mcnemar's Test P-Value : <0.0000000000000002
##
##             Sensitivity : 0.77333
##             Specificity : 0.77778
##          Pos Pred Value : 0.15183
##          Neg Pred Value : 0.98523
##              Prevalence : 0.04892
##          Detection Rate : 0.03783
##    Detection Prevalence : 0.24918
##       Balanced Accuracy : 0.77556
##
##        'Positive' Class : 1
##
```

Table 2.2: Reduced LR - Initial Four Fold plot

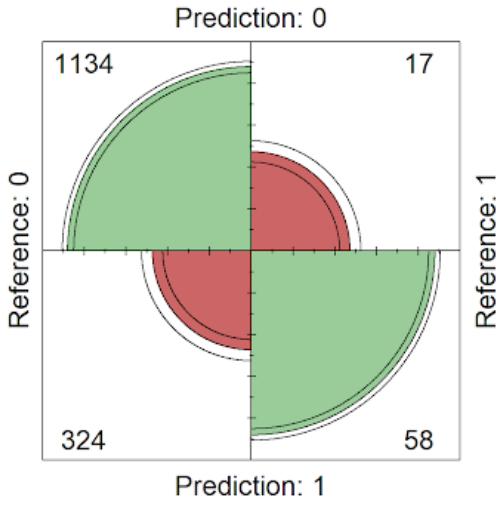
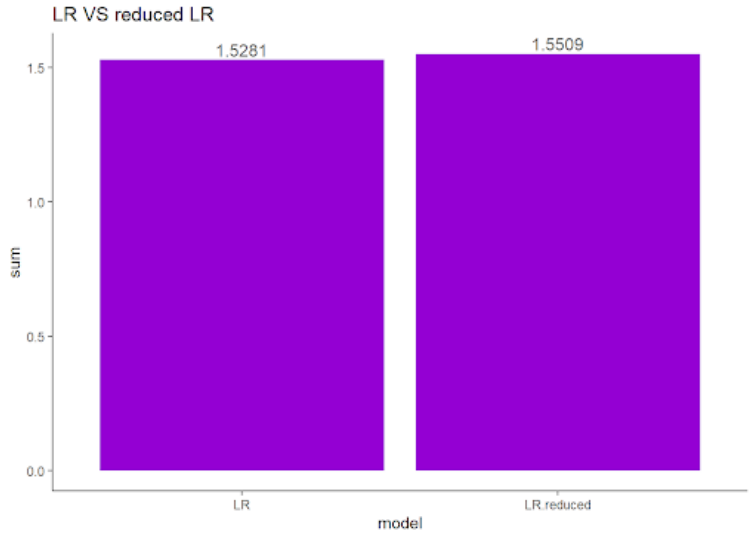


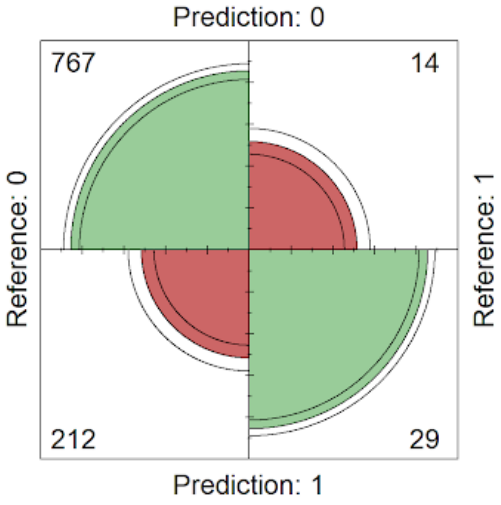
Table 3: Best model - LR vs LR reduced on validation set



Graph 1.1: Test set - Confusion matrix for LR

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction   0   1
##          0 767  14
##          1 212  29
##
##                Accuracy : 0.7789
##                  95% CI : (0.7521, 0.804)
##     No Information Rate : 0.9579
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.143
##
##  Mcnemar's Test P-Value : <0.0000000000000002
##
##             Sensitivity : 0.67442
##             Specificity : 0.78345
##          Pos Pred Value : 0.12033
##          Neg Pred Value : 0.98207
##              Prevalence : 0.04207
##          Detection Rate : 0.02838
##    Detection Prevalence : 0.23581
##       Balanced Accuracy : 0.72894
##
##        'Positive' Class : 1
##
```

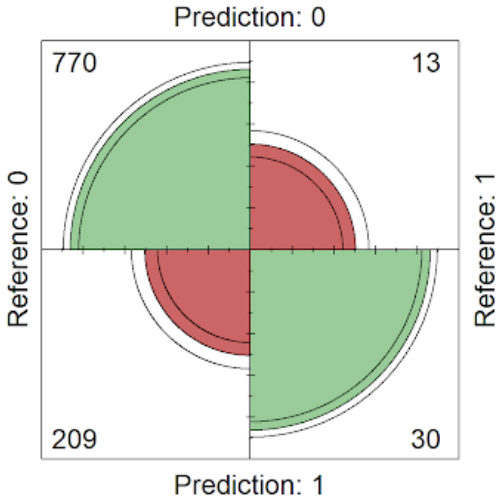
Graph 1.2: Test set - Four Fold plot for LR



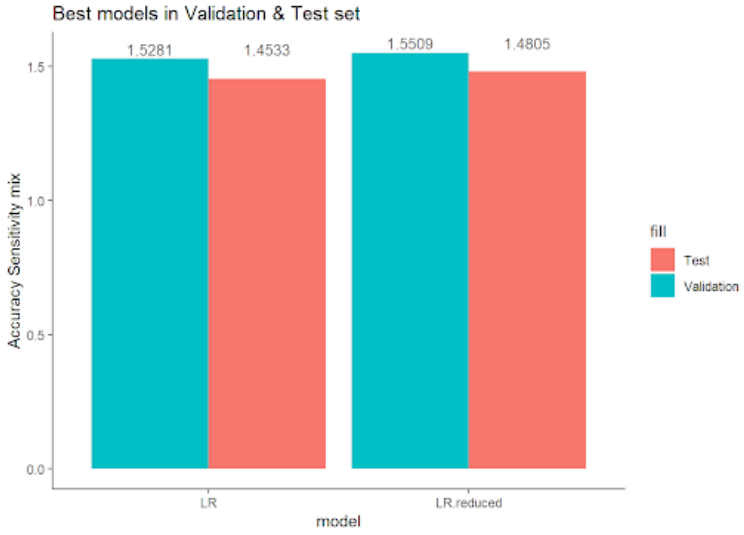
Graph 2.1: Test set - Confusion matrix for Reduced LR

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 770  13
##          1 209  30
##
##                Accuracy : 0.7828
##                  95% CI : (0.7562, 0.8077)
##     No Information Rate : 0.9579
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1523
##
##  Mcnemar's Test P-Value : <0.0000000000000002
##
##             Sensitivity : 0.69767
##             Specificity : 0.78652
##          Pos Pred Value : 0.12552
##          Neg Pred Value : 0.98340
##              Prevalence : 0.04207
##          Detection Rate : 0.02935
##    Detection Prevalence : 0.23386
##       Balanced Accuracy : 0.74210
##
##        'Positive' Class : 1
##
```

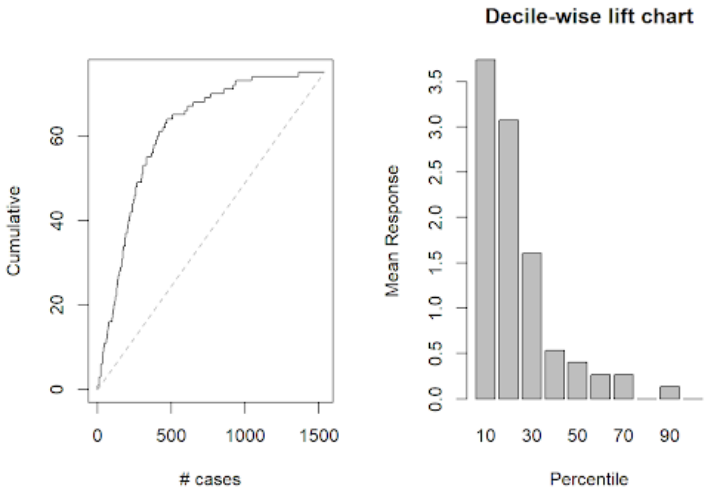
Graph 2.2: Test set - Four Fold plot for Reduced LR



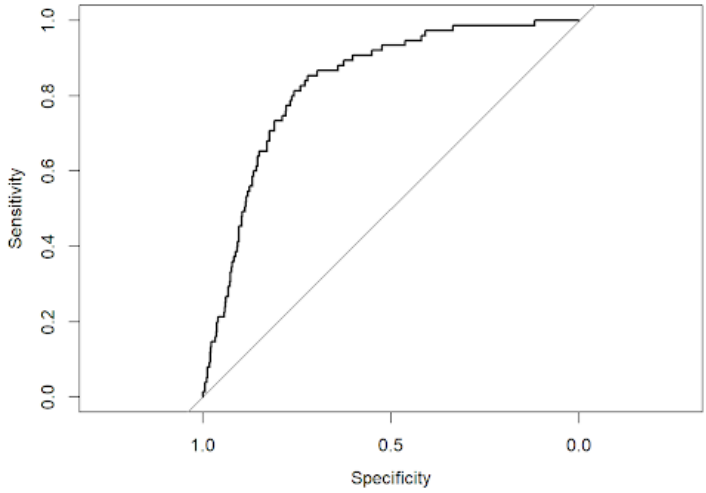
Graph 3: Test set - Model Comparison (LR + Reduced)



Graph 4.1: Best model - Gain & Decile chart

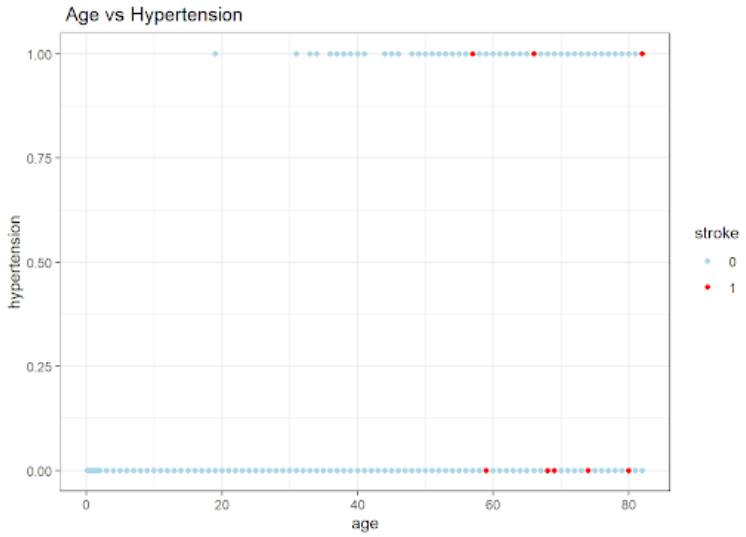


Graph 4.2: Best model - ROC curve



Part 7 - Linear Discriminant Analysis

Graph 1: LDA - Scatterplot of Age vs Hypertension



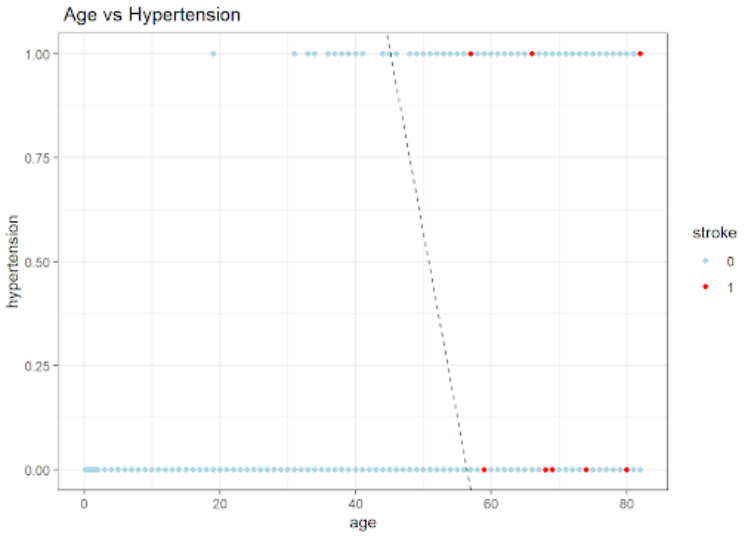

```
## Call:
## lda(stroke ~ age + hypertension, data = valid.lr)
##
## Prior probabilities of groups:
##         0         1
## 0.95107632 0.04892368
##
## Group means:
##        age hypertension
## 0 41.88702   0.08367627
## 1 67.50667   0.21333333
##
## Coefficients of linear discriminants:
##                    LD1
## age          0.04368914
## hypertension 0.49233089
```

Table 1.1: LDA - Output 1

```
##
## Linear Discriminant Analysis
## ----------------------------------------------
## $functions        discrimination functions
## $confusion        confusion matrix
## $scores           discriminant scores
## $classification   assigned class
## $error_rate       error rate
## ----------------------------------------------
##
## $functions
##                0       1
## constant    -1.9871  -7.7924
## age          0.0901   0.1418
## hypertension -0.7375  -0.1550
##
##
## $confusion
##         predicted
## original    0    1
##        0 1458    0
##        1   75    0
##
##
## $error_rate
## [1] 0.04892368
##
##
## $scores
##              0          1
## 4119  1.7889121 -1.9771291
## 1380  1.6086201 -2.2680013
## 1050  0.8874523 -3.3954002
## 1708  0.1662844 -4.5301790
## 1087  2.6903719 -0.5587681
## 2566  2.9600899 -0.1332598
## ...
##
## $classification
## [1] 0 0 0 0 0 0
## Levels: 0 1
## ...
```

Table 1.2: LDA - Output 2

Graph 2: LDA - Scatterplot with DA line



References

Lahera, German. (2019). Unbalanced datasets and what to do about them. https://medium.com/strands-tech-corner/unbalanced-datasets-what-to-do-144e0552d9cd.

Ltd, MedCalc Software. (2021). Roc curve analysis. https://www.medcalc.org/manual/roc-curves.php.

M, Rekha. (2019). Correlation and collinearity - how they can make or break a model. https://blog.clairvoyantsoft.com/correlation-and-collinearity-how-they-can-make-or-break-a-model-9135fbe6936a.

Mayo-Clinic, Staff. (2021). Stroke - symptoms and causes. https://www.mayoclinic.org/diseases-conditions/stroke/symptoms-causes/syc-20350113.

Office, Office fédéral des routes OFROU. (2018). 5,8 millions de personnes possèdent le permis voiture en suisse. https://www.astra.admin.ch/astra/fr/home/documentation/communiques-de-presse/anzeige-meldungen.msg-id-71190.html.

Organization, World Stroke. (2021). Why stroke matters. https://www.world-stroke.org/world-stroke-day-campaign/why-stroke-matters.

Shmueli, Galit and Peter C. Bruce, Nitin R. Patel Kenneth C. Lichtendahl Jr. Inbal Yahav. (2018). Data mining for business analytics: Concepts, techniques, and applications in r. **7**.

Creating Value through Data Mining

Veronica Rowe and Malou-Tinette Kouango

Predicting stroke from "Stroke Prediction" dataset

15th November 2021

Overview

Our project's objective is to predict the propensity to get a stroke based on key attributes of a patient, using :

- Exploratory Data Analysis and Data Visualisation
- Data Processing and Standardization
- Classification and Predictive Modelling methods

Goals

1. Which variables have the highest impact on the liability to get a stroke ?
2. Can we predict the likelihood of a patient getting a stroke with the patient's characteristics ?
3. From the data, which strategy/what policies would be the most effective for a public health campaign ?

Specifications

The dataset consists in hospital records of 5110 patients with 12 attributes.

ID	Unique identifier
gender	Patient gender (Male, Female or Other)
age	Age of the patient
hypertension	Whether the patient has hypertension or not (0,1)
heart_disease	Whether the patient has a heart disease or not (0,1)
ever_married:	Whether the patient has ever been married (Yes, No)
work_type	Patient profession type (children, Government job, Private job, Self-employed or Never worked)
Residence_type	Patient type of residence (Rural or Urban)
avg_glucose_level	Average glucose level in blood
bmi	Body mass index
smoking_status	Whether the patient is/was a smoker or not (formerly smoked, never smoked, smokes or Unknown*)
stroke	Whether the patient had a stroke or not (0,1)

*Note: "Unknown" in smoking_status means that the information is unavailable for this patient

Link to dataset : https://www.kaggle.com/fedesoriano/stroke-prediction-dataset

Link to GitHub: MalouTK/cvtdm: CVTDM Project (github.com)